EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Enterprise Computing
Sysplex

Prof. Dr.-Ing. Wilhelm G. Spruth
Dipl. Inf. Gerald Kreißig

WS2016/17

Sysplex Teil 1

Mehrfachrechner

Mehrfachrechner

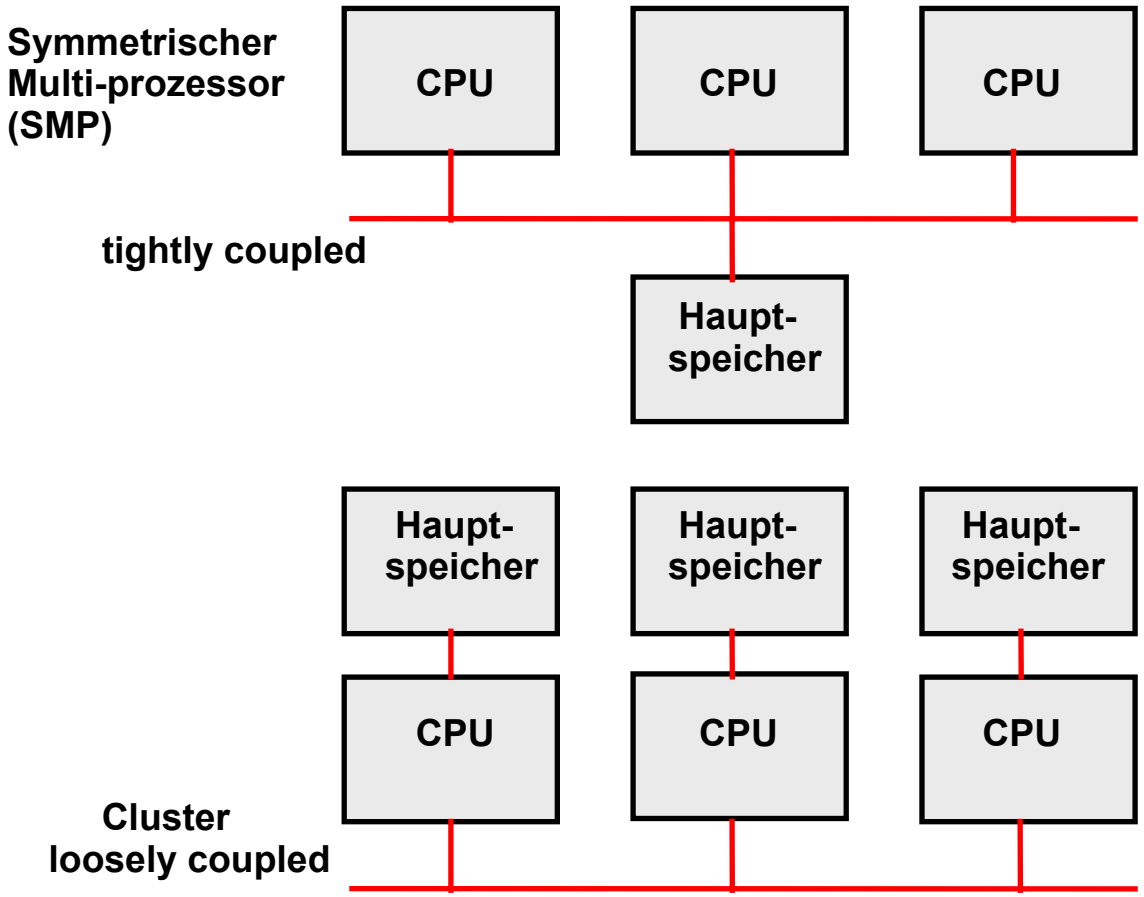
Eine einzelne CPU hat eine nur begrenzte Rechenleistung. Die Forderung nach mehr Rechenleistung führte dazu, mehrere Prozessoren zu koppeln.

Wir unterscheiden zwischen Parallelrechnern und Mehrfachrechnern. Bei Parallelrechnern wird ein einziger Prozess auf vielen CPUs gleichzeitig ausgeführt. Beispiele für solche Prozesse sind Wetter- und Klimamodelle, der Automobil Crash Test, Kosmologische Simulationen oder Seismische Exploration.

Bei Mehrfachrechnern laufen zahlreiche Prozesse oder Threads gleichzeitig auf eben so vielen CPUs. In Systemen mit mehr als einer CPU teilt das Betriebssystem die einzelnen Prozesse oder Threads auf die verschiedenen Prozessoren auf und sorgt damit für eine höhere Leistungsfähigkeit.

Mainframe Systeme erfordern meistens die Leistung von mehr als einer CPU, arbeiten aber fast immer als Mehrfachrechner. Ausnahmen sind z.B. Data Mining Anwendungen.

Taxonomie von Mehrfachrechnern

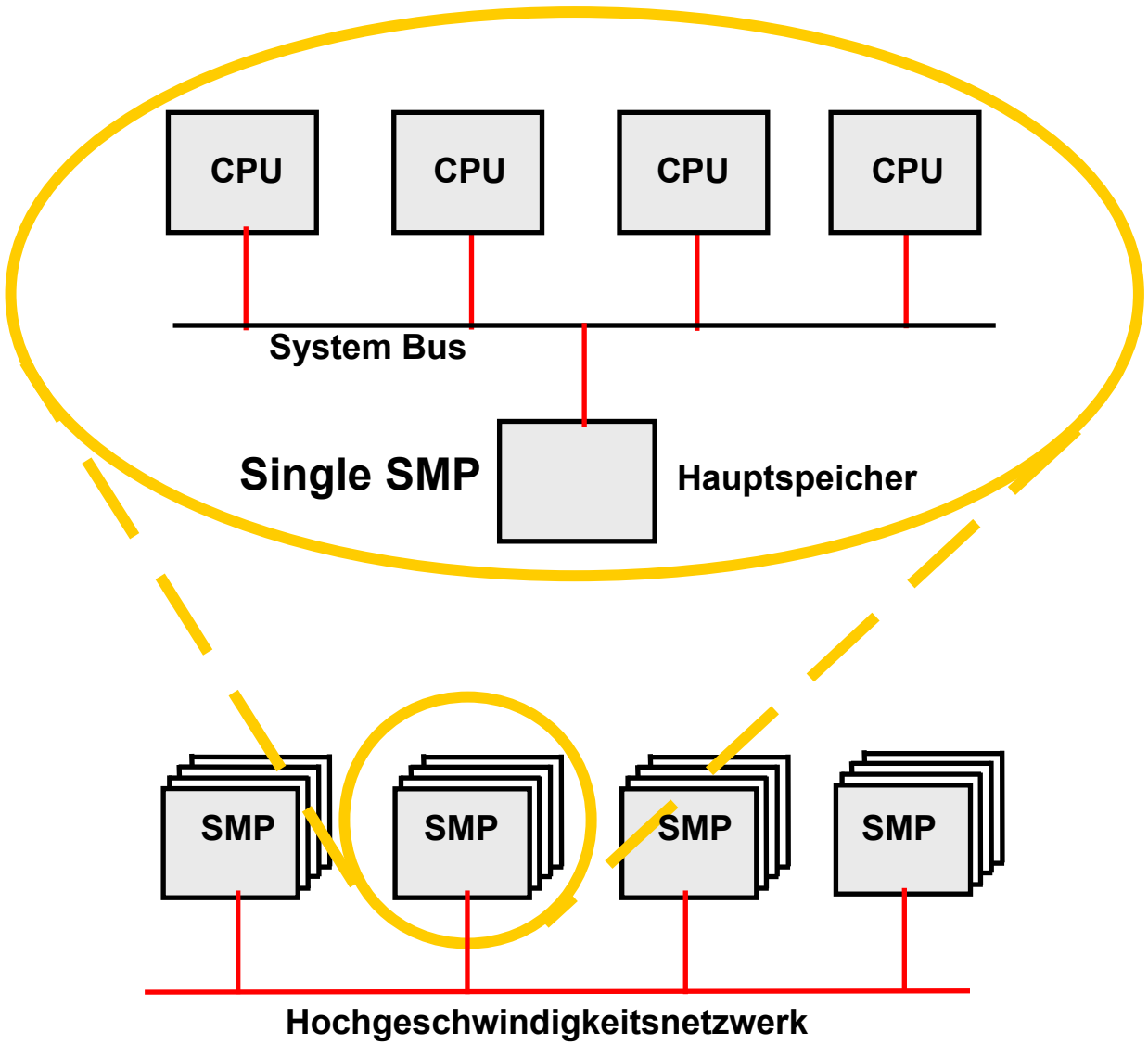


Bei Mehrfachrechnern unterscheiden wir 2 Grundtypen:

- Ein **Symmetrischer Multiprozessor (SMP)** besteht aus mehreren CPUs (über 100 bei einem zEC12 Mainframe), die alle auf einen gemeinsamen Hauptspeicher zugreifen. In dem Hauptspeicher befindet sich eine einzige Instanz des Betriebssystems. Er wird auch als **eng gekoppelter (tightly coupled) Mehrfachrechner** bezeichnet.
- Ein **Cluster (loosely coupled Mehrfachrechner)** besteht aus mehreren Rechnern, von denen jeder seinen eigenen Hauptspeicher und seine eigene Instanz eines Betriebssystems hat.

Cluster

bestehend aus mehreren SMPs



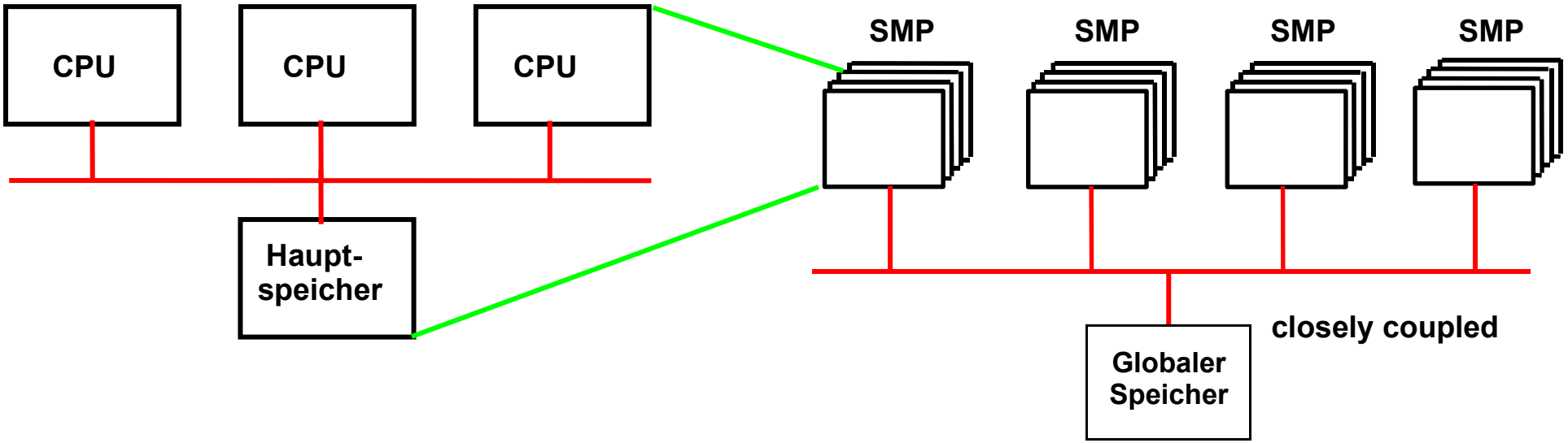
Häufig (heute fast immer) sind die Elemente eines Clusters nicht einzelne CPUs, sondern SMPs, die als Knoten (und in der IBM Terminologie als „Systeme“) bezeichnet werden.

Jeder Knoten ist ein SMP. Er besteht aus mehreren CPUs, die auf einen gemeinsamen Hauptspeicher mit einer einzigen Instanz des Betriebssystems zugreifen.

Die Knoten sind über ein Hochgeschwindigkeitsnetzwerk miteinander verbunden, das in der Regel als Crossbar Switch implementiert wird.

Die Großrechner von HP, IBM und Sun haben diese Struktur.

Ein SMP wird als tightly coupled, ein Cluster als loosely coupled Multiprozessor bezeichnet.



Ein Cluster, bei dem die Knoten aus einzelnen CPUs oder aus SMPs bestehen, kann durch einen globalen Speicher erweitert werden. Diese Konfiguration wird als **nahe gekoppelt (closely coupled)** bezeichnet.

Crossbar Switch

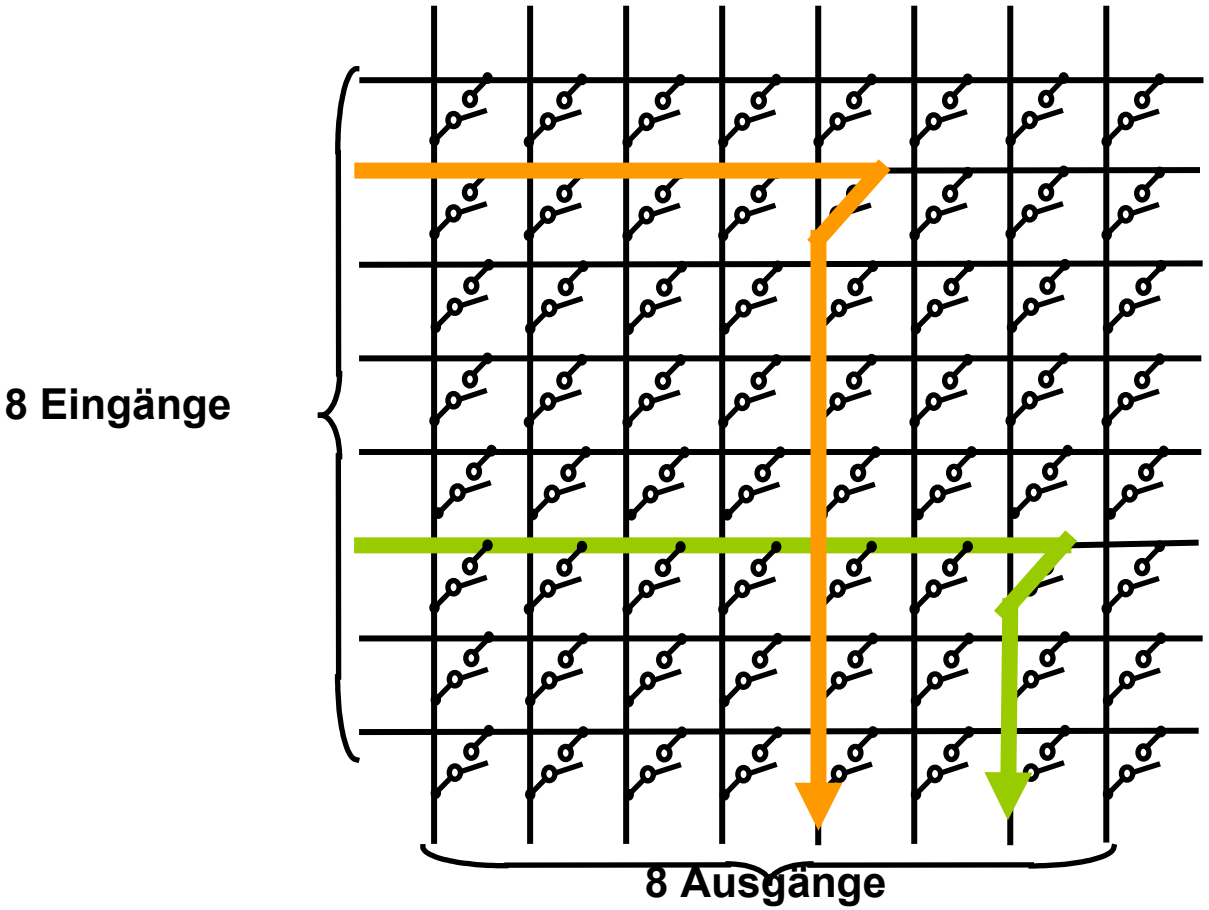
Die CPUs eines Mehrfachrechners sind über ein Verbindungsnetzwerk miteinander verbunden. Dies kann ein leistungsfähiger Bus sein, z.B. der PCI Bus. Ein Bus hat aber nur eine begrenzte Datenrate. Deshalb setzen viele Implementierungen statt dessen einen Kreuzschienenverteiler (Crossbar Switch, Crossbar Matrix Switch) als Verbindungsnetzwerk ein, der die gleichzeitige Verbindung mehrerer Eingänge mit mehreren Ausgängen ermöglicht.

Mit derartigen Switches können fast beliebige Datenraten erreicht werden. In der folgenden Abbildung ist als Beispiel ein Switch mit 8 Eingängen und 8 Ausgängen gezeigt, der gleichzeitig 8 parallele Verbindungen ermöglicht. In jedem Augenblick kann durch entsprechende Steuerung jeder Eingang mit jedem Ausgang verbinden sein.

Die Personalisierung der 8 x 8 = 64 Transistor-Switche erfolgt in einfachsten Fall durch einen 8 x 8 Bit Speicher, der durch einen eigenen Mikroprocessor angesteuert wird. Durch Änderung des Speicherinhalts können die Verbindungen dynamisch geändert werden.

Zur Erhöhung der Brandbeite lassen sich 8 oder 32 derartige Switche übereinanderstapeln, um einen 8 oder 32 Bit breiten Bus für jeden Eingang/Ausgang zu implementieren.

Crossbar Switch Silizium Chips mit je 256 Ein/Ausgängen lassen sich in der heutigen Silzium Technik kostengünstig herstellen.



Beispiel: 8 x 8 Crossbar Switch

Unix Großrechner

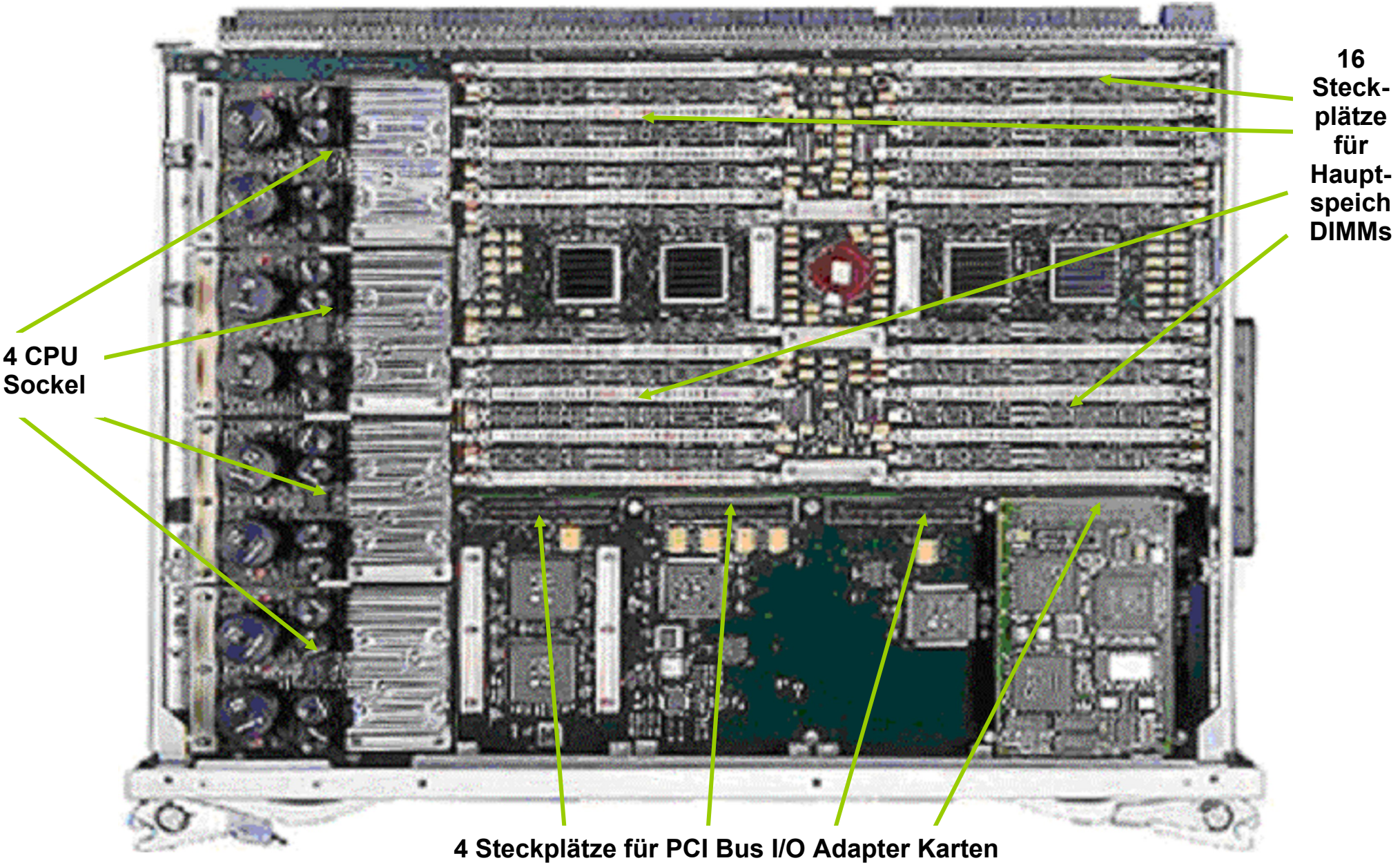
Die folgenden Abbildungen zeigen Beispiele von Unix basierten Großrechnern der Firmen Sun und Hewlett Packard. Das erste Beispiel betrifft den E 15000 Rechners der Firma Sun.

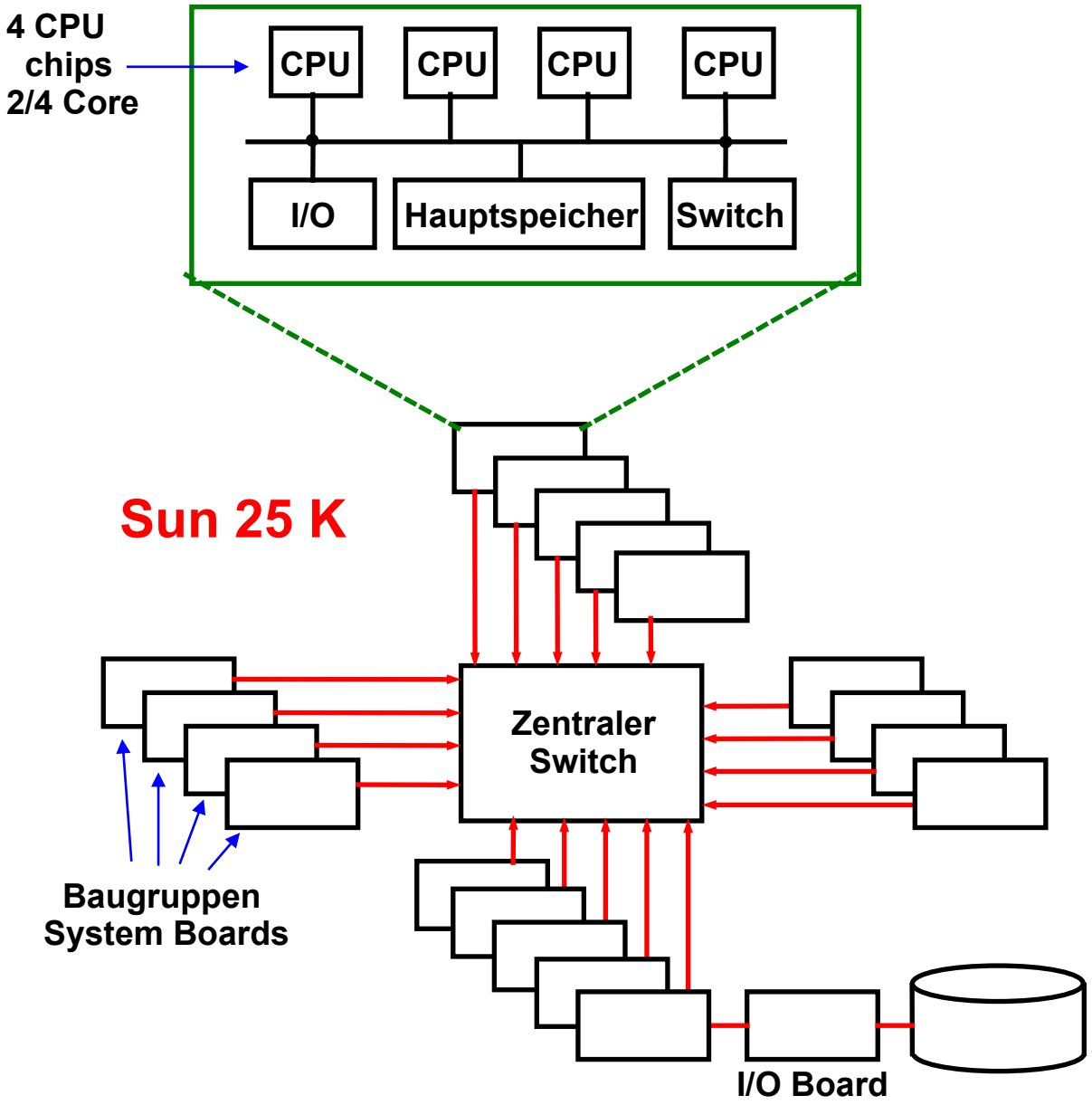
Zentralstück des Rechners ist ein Crossbar Switch, der sich auf einem Motherboard zusammen mit 16 Steckplätzen für sog. „System Boards“ befindet. Die folgende Abbildung zeigt ein derartiges System Board.

Auf dem System Board befinden sich 4 CPU Chip Sockel. Zu sehen sind die Kühlkörper. CPU Chips waren früher Single Core, heute sind sie in der Regel aber Dual-Core, Quad-Core oder mehr. Weiterhin befinden sich auf dem System Board 16 Steckplätze für Hauptspeicher DIMMs.

Zusätzlich enthält das System Board 4 Steckplätze für sog. Daughter Cards. Es stehen eine Reihe unterschiedlicher Arten von Daughter Cards zur Verfügung, aber die allermeisten Steckplätze werden von I/O Adapter Karten eingenommen, die z.B. mit einem SCSI Kabel eine Verbindung zu Plattenspeichern ermöglichen.

Sun Fire 15000 System Board





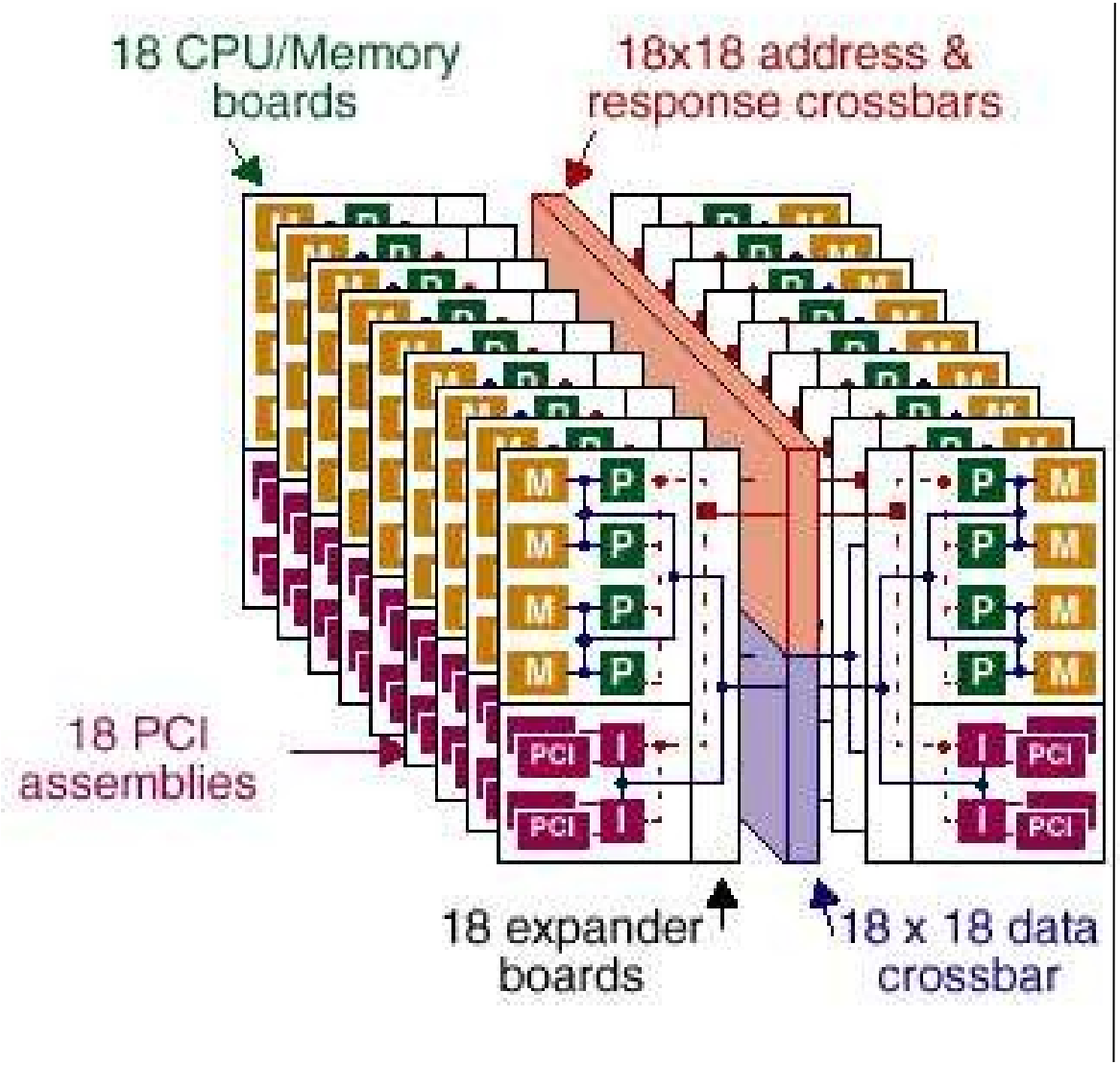
Ein Sun 15K oder HP Superdome Großrechner enthält bis zu 18 Printed Circuit Board (PCB) Baugruppen (System Boards), jedes mit 4 CPU Chip Sockeln und bis zu 128 GByte Hauptspeicher, einem Anschluss an einen zentralen Switch sowie I/O Adapter auf jedem System Board.

Die I/O Adapter sind mit PCI Bus Kabeln mit einer Reihe von I/O Cages verbunden, in denen sich Steckkarten für den Anschluss von I/O Geräten, besonders Festplattenspeichern, befinden.

Die CPUs eines jeden System Boards können nicht nur auf den eigenen Hauptspeicher, sondern auch auf den Hauptspeicher eines jeden anderen System Boards zugreifen. Hiermit wird eine „Non-Uniform Memory Architecture“ (NUMA) verwirklicht.

Eine moderne Form eines System Boards wird als „Blade“ bezeichnet, und befindet sich in vielen Low End Produkten der Firma Sun.

Sun Fire 15000



Die 18 System Boards befinden sich in Steckplätzen auf einem Crossbar Board. Dieses enthält eine Reihe von Crossbar Chips, welche alle System Boards miteinander verbindet. Dies sind spezifisch ein 18 x 18 Daten Crossbar Chip, ein 18 x 18 Response Crossbar Chip und ein 18 x 18 Adressen Crossbar Chip.

Das Crossbar Board enthält eine Reihe von (elektronischen) Schaltern. Mit diesen lassen sich die System Bards in mehrere Gruppen aufteilen und die Gruppen voneinander isolieren.

Da ein Unix SMP mit 64, 128 oder 256 CPUs bei Transaktions- und Datenbank-anwendungen nicht mehr skaliert, kann der Rechner mit Hilfe der Schalter in mehrere voneinander isolierte SMPs aufgeteilt werden.

Eine solche Aufteilung wird als „Harte Partitionierung“ bezeichnet.

Hewlett Packard Superdome

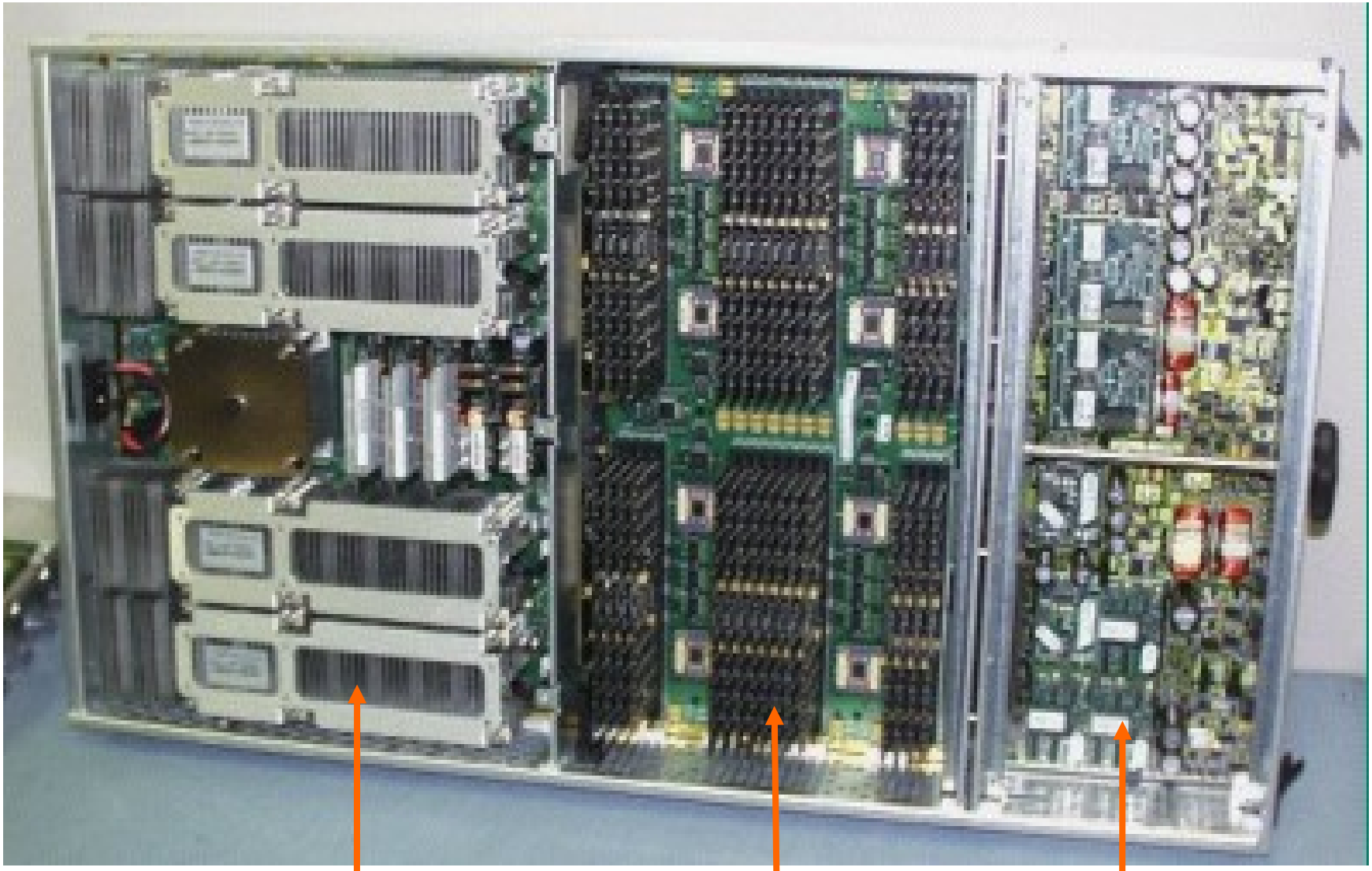
Der Hewlett Packard (HP) Superdome Rechner ist ähnlich aufgebaut wie der Sun Fire 15K bzw. Sun M9000 Rechner. Das System Board wird als Cell Board bezeichnet und hat 4 Sockel für 4 Itanium (Tukwila) CPU Chips mit je 4 CPU Cores. Neben den CPU Chips befinden sich auf dem Cell Board Sockel für Hauptspeicherp DIMMs sowie Steckplätze für PCI Bus I/O Adapter Karten. Wie beim Sun Fire 15K sind die Cell Boards über einen zentralen Cross Bar Matrix Switch miteinander verbunden.

Auch die Firma Hewlett verwendet eine vereinfachte Form eines Cell Boards (als „Blade“ bezeichnet) in vielen Low End Produkten des Unternehmens. Cell Boards haben gegenüber Blades zahlreiche zusätzliche Funktionen. Beispiele sind:

- Verbindung über einen zentralen Switch
- NUMA Fähigkeit
- Partitionsmöglichkeiten (siehe nächstes Thema)
- Zusätzliche unterstützende Hardware für das HP-UX Betriebssystem
- Verbesserte I/O Einrichtungen, höhere I/O Kapazität
- Einrichtungen für eine zentrale Administration
- Zusätzliche Verbesserungen für Ausfallsicherheit, Zuverlässigkeit und Verfügbarkeit
- .....

IBM bietet als Alternative zu den Sun M9000 oder HP Superdome Systemen mit den Solaris oder HP-UX Betriebssystemen das ähnliche Produkt „System p“ mit den Betriebssystemen AIX und i5/OS an. Aus Platzgründen wird auf diese Entwicklung nicht näher eingegangen. AIX ist ein besonders funktionsreiches Unix Betriebssystem.

Hewlett-Packard Superdome Cell Board



4 Itanium 2 CPUs 1,73 GHz

64 Gbyte Hauptspeicher

E/A Bus Anschlüsse

http://www.serverworldmagazine.com/webpapers/2001/05_hpsuperdome.shtml

System Processing Complex
Sysplex

Der **Sysplex (SYStem Processing compLEX)** ist eine 1990 für Mainframes eingeführte lose Rechnerkopplung von IBM Mainframes. Oft wird diese einfache Form des Clusters auch Base Sysplex genannt.

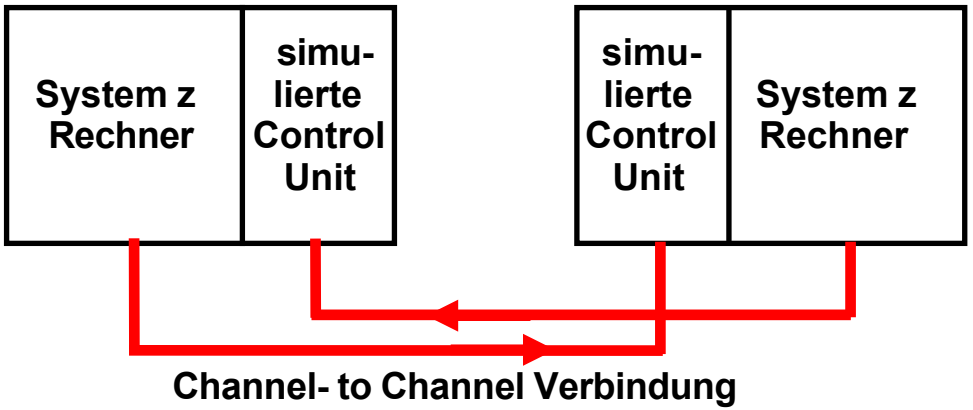
Der **Parallel Sysplex** ist eine Weiterentwicklung des Base Sysplex, dessen Einführung 1994 erfolgte. Hierbei handelt es sich um eine **nahe Rechnerkopplung (closely coupled)**. Da der Base Sysplex praktisch ausgestorben ist, bezeichnet ein Sysplex in der Umgangssprache hetzutage fast immer einen Parallel Sysplex.

Beim Parallel Sysplex übernimmt die **Coupling Facility (CF)** die Funktion eines globalen Arbeitsspeichers. Innerhalb der CF können drei verschiedene Datenstrukturtypen erzeugt und verwaltet werden. Dies sind Sperr- (Lock), Listen- und Cache-Strukturen, die u.a. genutzt werden, um Konkurrenzsituationen beim Zugriff auf Rescourcen zu verwalten. Die Hardware der Coupling Facility besteht aus einem regulären Mainframe Rechner, auf dem der „Coupling Facility Control Code“ (CFCC) an Stelle eines Betriebssystems läuft.

Nutzen Subsysteme wie CICS, DB2 oder IMS die durch den Parallel Sysplex bereitgestellten Einrichtungen, so können sie sich gegenüber dem Anwender als eine einzige Anwendung präsentieren. Man spricht dann auch von einem Single System Image (SSI).

Literatur: Wilhelm G. Spruth, Erhard Rahm, Sysplex-Cluster Technologien für Hochleistungs-Datenbanken. Datenbank-Spektrum, Heft 3, 2002, S. 16-26. http://www-ti.informatik.uni-tuebingen.de/~spruth/Mirror/Sysplex3.pdf

Cross-System Coupling Facility (XCF)



Die **Cross-System Coupling Facility (XCF)** ist eine Komponente des z/OS Kernels.

Sie verwendet das **Channel To Channel (CTC)** Protokoll und stellt die Coupling Services bereit, mit denen z/OS Systeme innerhalb eines Sysplex miteinander kommunizieren.

Für eine CTC Verbindung stellt ein System z oder S/390 Rechner (als Slave bezeichnet) eine simulierte Control Unit zur Verfügung. An diese ist ein anderer Rechner (Master) über eine normale FICON Glasfaserverbindung angeschlossen. Der Master kommuniziert mit dem Slave wie mit einer normalen I/O Einheit.

Die CTC Verbindung ist unidirectional. Aus Symmetriegründen werden CTC Verbindungn normalerweise paarweise eingerichtet.

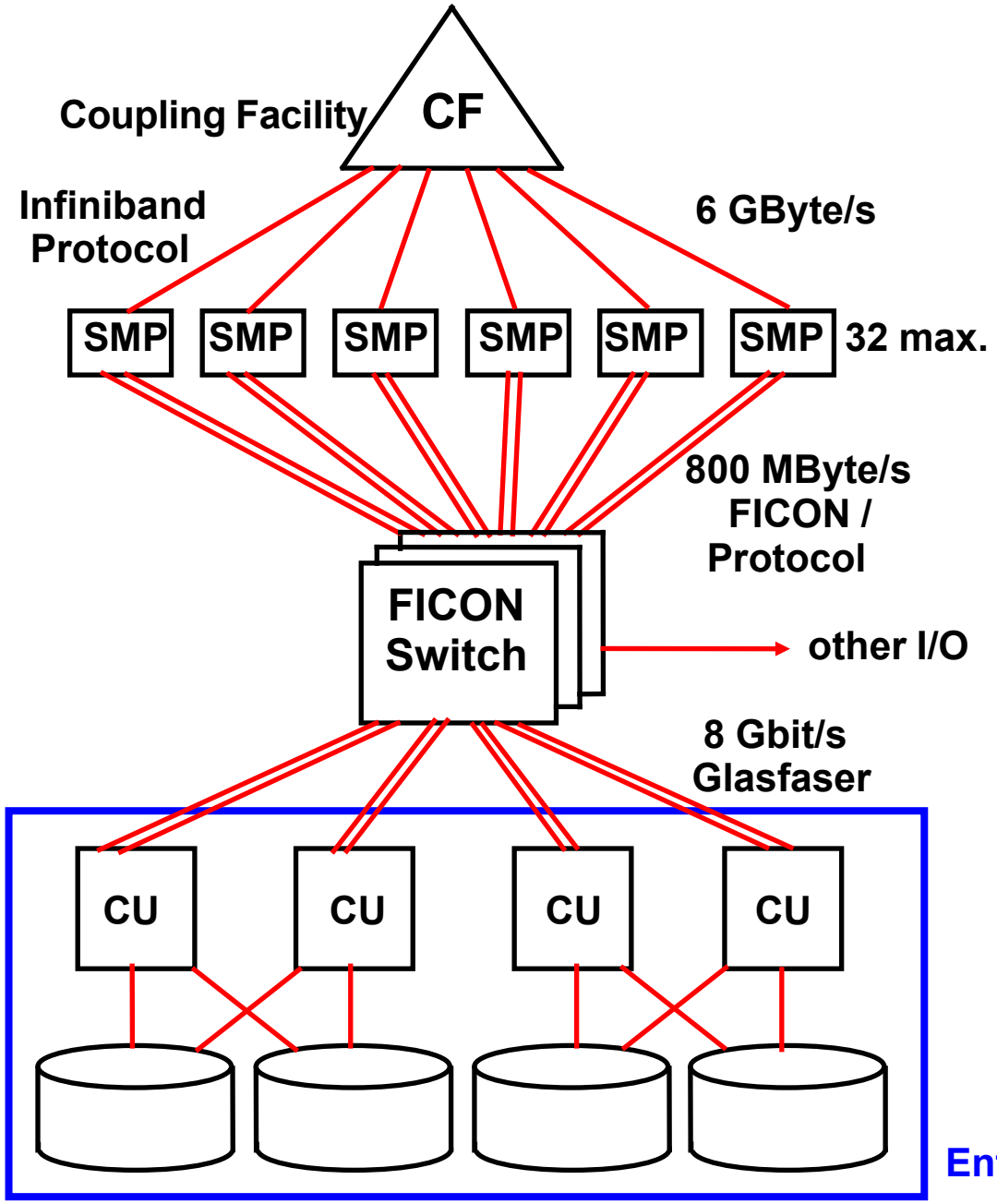
Mittels CTC und XCF können die Rechner eines Sysplex miteinander kommunizieren. Dies geschieht über FICON Directors, und nutzt das gleiche FICON Netzwerk, was auch für die Verbindung der Rechner mit Enterprise Storage Servern und Magnetbändern benutzt wird.

Parallel Sysplex
Sysplex mit Coupling Facility

Die folgende Abbildung zeigt den Aufbau der Parallel Sysplex-Architektur. Während die vorgestellten Cluster von Sun und HP eine lose Kopplung verwenden, handelt es sich beim Parallel Sysplex um eine **nahe gekoppelt (closely coupled)** Cluster-Architektur. Der Sysplex besteht aus den **Prozessor-Knoten** (fast immer SMPs), gemeinsamen Plattenspeichern (Shared Storage Devices), Netzwerk-Controllern und den Kern-Cluster-Technologie-Komponenten. Letztere umfassen den (die) als „FICON Director“ bezeichneten Switch(es), den Sysplex-Timer und die „Coupling Facility“ (CF). Die Coupling Facility enthält den für die nahe Kopplung charakteristischen globalen Speicher zur Realisierung globaler Kontrollaufgaben und ist von allen Knoten schnell zugreifbar.

Zu den CPUs kommen noch weitere Ein-/Ausgabe-Prozessoren (System Assist-Prozessoren, SAPs). Heutige Installationen haben bis zu 200 CPUs. Die Knoten müssen nicht homogen sein, d.h. es können unterschiedliche Mainframe Modelle eingesetzt werden.

Zu den Sysplex-Komponenten werden spezifische Maschinenbefehle sowie Betriebssystemdienste zur Verfügung gestellt, mit denen eine effiziente Durchführung der Cluster-Aufgaben für alle Knoten erreicht wird, insbesondere zur Kommunikation, Ein/Ausgabe, sowie für globale Steuerungsaufgaben wie Synchronisation, Kohärenzkontrolle und Lastbalancierung. Diese Dienste werden in allgemeiner Form realisiert und von unterschiedlichen Software-Subsystemen, insbesondere Web Application Server, Datenbanksystemen und Transaktionsservern für den Cluster-Einsatz verwendet. IBM hat dazu die wichtigsten Subsysteme an das Arbeiten innerhalb eines Sysplex angepasst. Eine Anpassung der Anwendungen an den Sysplex ist in der Regel nicht erforderlich und i.A. auch nicht möglich.



Parallel Sysplex mit Coupling Facility

Ein Sysplex besteht aus bis zu 32 System z Rechnern (Knoten), 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 Rechner, auf dem „Coupling Facility Code“ an Stelle eines Betriebssystems läuft.

Weiterhin sind 1 oder 2 Zeitgeber (Sysplex Timer) vorhanden. Bei den heutigen Rechnern ist die Zeitgeberfunktion in die CPU Chips integriert.

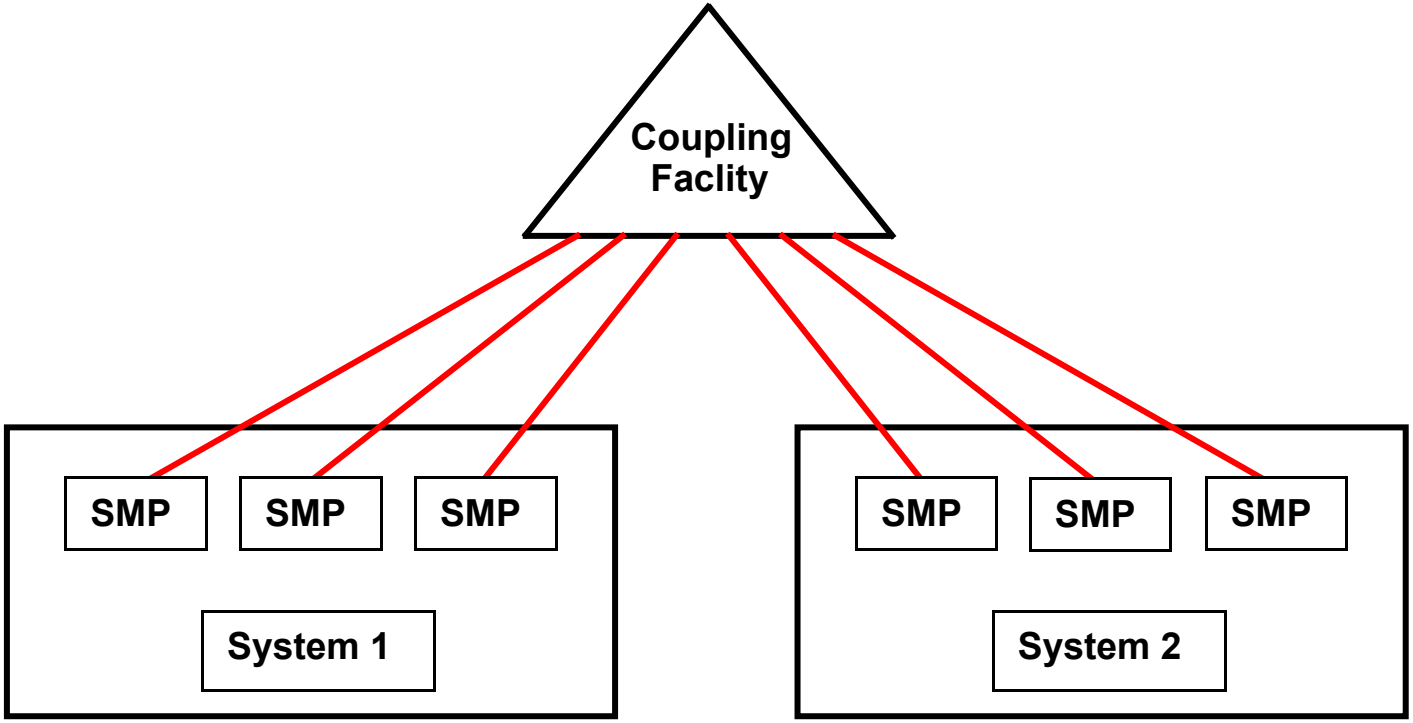
Enterprise Storage Server

Jeder Knoten kann mit bis zu 256 „Kanälen“ mit der Außenwelt verbunden werden, mit bis zu 800 MByte/s pro Kanal. Die Verbindung der Systeme sowohl untereinander als auch zu den Festplattenspeichern erfolgt über ein oder mehrere FICON (FIbre CONnection)-Switches. Da jedes System 256 Ein-/Ausgabe-Kanäle anbinden kann, hat diese Verbindungsstruktur die Aufgabe, bei 32 Systemen maximal 256 * 32 = 2^{13} Kanäle zu verwalten. Damit ist jeder Knoten in der Lage, auf alle Plattenspeicher und alle anderen Knoten des Sysplex direkt zuzugreifen (Shared Disk-Modell). Die Steuereinheiten der Plattenspeicher (Control Units) implementieren eine Storage Server- und SAN-Funktionalität.

Die von IBM entwickelte FICON-I/O-Architektur basiert auf dem Kanal-Subsystem (channel subsystem) der Mainframe I/O-Architektur. Dieses integriert System Assist Prozessoren (SAPs), Kanäle sowie die „Staging Hardware“. Die SAPs führen die Kommunikation zwischen den Knoten und den Kanälen durch. Diese führen zur Datenübertragung ein Kanalprogramm aus, wobei über die Steuereinheiten (Control Units) auf die Plattenspeicher zugegriffen wird. Die Staging Hardware stellt die Kommunikationspfade zwischen den I/O-Prozessoren, den Kanälen und dem Rest des Systems zur Verfügung.

In der FICON-Architektur bildet der FICON-Switch (FICON Director) die Kerneinheit. Dies implementiert eine Switched Point-to-Point-Topologie für System z I/O-Kanäle und Steuereinheiten. Ein FICON-Switch kann bis zu 60 Kanäle und Steuereinheiten dynamisch und nicht-blockierend (Crossbar Switch) über seine Ports miteinander verschalten. Normalerweise werden eine Reihe von FICON-Switches parallel genutzt. Entfernungen von bis zu 3 km für optische Übertragungen sind möglich. Die zulässigen Entfernungen erhöhen sich beim Einsatz einer sogenannten Extended Distance Laser Link-Einrichtung auf 20, 40 oder 60 km. Dies ist für Unternehmen wichtig, die aus Katastrophenschutz Gründen zwei getrennte Rechenzentren betreiben.

Die Kommunikation zwischen den Knoten und den Plattenspeichern erfolgt über das FICON-Protokoll. Zur Kommunikation der Knoten untereinander wird das „Channel-to-Channel“ (CTC)–Protokoll eingesetzt, normalerweise in einer Full-Duplex-Anordnung. Hierbei betrachtet jeder sendende Knoten den Empfänger als eine Ein-/Ausgabe-Einheit. Der Empfänger simuliert für diesen Zweck einen eigenen Typ einer Mainframe-Steuereinheit. Das z/OS-Betriebssystem stellt einen Basisdienst, die „Cross System Coupling Facility“ (XCF) zur Verfügung, um über eine CTC-Verbindung eine Kommunikation mit einer anderen z/OS-Instanz innerhalb eines Sysplex zu bewerkstelligen. Dieser Dienst wird wiederum von Subsystemen wie den Datenbanksystemen DB2 und IMS für den Nachrichtenaustausch eingesetzt.



In einem Sysplex werden bis zu 32 Knoten unterstützt. Jeder Knoten stellt einen SMP dar und enthält maximal 101 CPUs, insgesamt also maximal 3232 CPUs. Dieser Wert kann in der Praxis nicht erreicht werden, weil SMPs in der Regel nicht bis zu 101 CPUs skalieren. Eine große Sysplex Installation besteht deshalb in der Regel aus mehreren physischen Rechnern (Systeme in der IBM Terminologie), von denen jeder mehrere SMPs (Knoten) enthält. Aus logischer Sicht spielt es dabei keine Rolle, ob zwei SMPs im gleichen oder in getrennten physischen Rechnern untergebracht sind.

Jeder Knoten ist über eine eigene Glasfaserverbindung (Coupling Link) mit der Coupling Facility verbunden.

Zur Erhöhung der Reliability und Availability werden in der Praxis fast immer 2 Coupling Facilities eingesetzt, von denen die zweite CF als Backup für die erste CF dient.

Parallel Sysplex Cluster Technology

Zu den Parallel Sysplex Cluster Technology Komponenten gehören:

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

Die Coupling Facility ermöglicht Data Sharing einschließlich Datenintegrität zwischen mehrfachen z/OS Servern.

Sysplex Teil 2

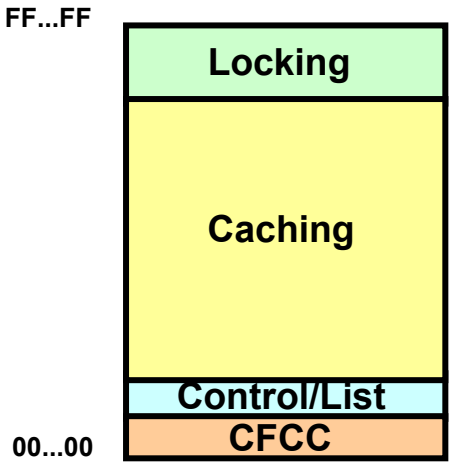
Coupling Facility

Coupling Facility (CF)

Die Coupling (CF) Facility ist in Wirklichkeit ein weiterer System z Rechner mit spezieller Software.
Die Aufgaben der CF sind:

- Locking
- Caching
- Control/List Structure Management

Der Hauptspeicher der Coupling Facility enthält einen als **Coupling Facility Control Code (CFCC)** bezeichnetes Betriebssystem, sowie Speicherbereiche für Locking, Caching und Control/List Strukturen.



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

Die wichtigste Aufgabe der Coupling Facility ist ein zentrales Lock Management für die angeschlossenen Knoten. Der zentrale Lock Manager des SAP System R/3 hat in Ansätzen eine ähnliche Funktionalität, jedoch ohne die CF spezifischen Eigenschaften.

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

CF Cache Cross-Invalidate (ungültig machen) Nachrichten gehen nur an die betroffenen Systeme.

Control und List Strukturen dienen der Sysplex-Cluster-weiten Verwaltung. Ein Beispiel ist ein den ganzen Cluster umfassendes RACF Sicherheits Subsystem. Ein weiteres Beispiel ist der bereits erwähnte WebSphere MQ Cluster (siehe Wintersemester MQSeries, Teil 4).

Eine interessante Überlegung: Das SAP System R/3 würde erheblich vom Vorhandensein einer Coupling Facility profitieren. Das Problem ist: SAP System R/3 ist als Multi-Plattform Software ausgelegt, lauffähig nicht nur unter z/OS, sondern auch unter den verschiedenen Unix Dialekten, Linux und Windows. Um eine CF nutzen zu können, wären identische Hardware Einrichtungen auf Sparc, Itanium, PowerPC und x86 Rechnern erforderlich.

Die Lauffähigkeit auf unterschiedlichen Platformen hat ihren Preis.

Coupling Facility Control Code

Eine Coupling Facility ist ein normaler System z Rechner, der **Coupling Facility Control Code (CFCC)** ausführt. Es handelt sich hierbei um ein normales hochspezialisiertes Betriebssystem mit einigen Besonderheiten.
CFCC ist jedoch Firmware. CFCC Code ist aber gleichzeitig normaler System z Maschinencode.

Dies bedeutet, eine Coupling Facility verhält sich wie eine Black Box. z/OS kann der CF eine Nachricht schicken, und die CF reagiert irgendwie darauf. Weder der z/OS Systemprogrammierer, und erst Recht nicht der Anwendungsprogrammierer sind in der Lage, die CF zu programmieren.

Es ist deshalb möglich, für Experimentierzwecke CFCC Code in einer virtuellen Maschine unter z/VM laufen zu lassen.

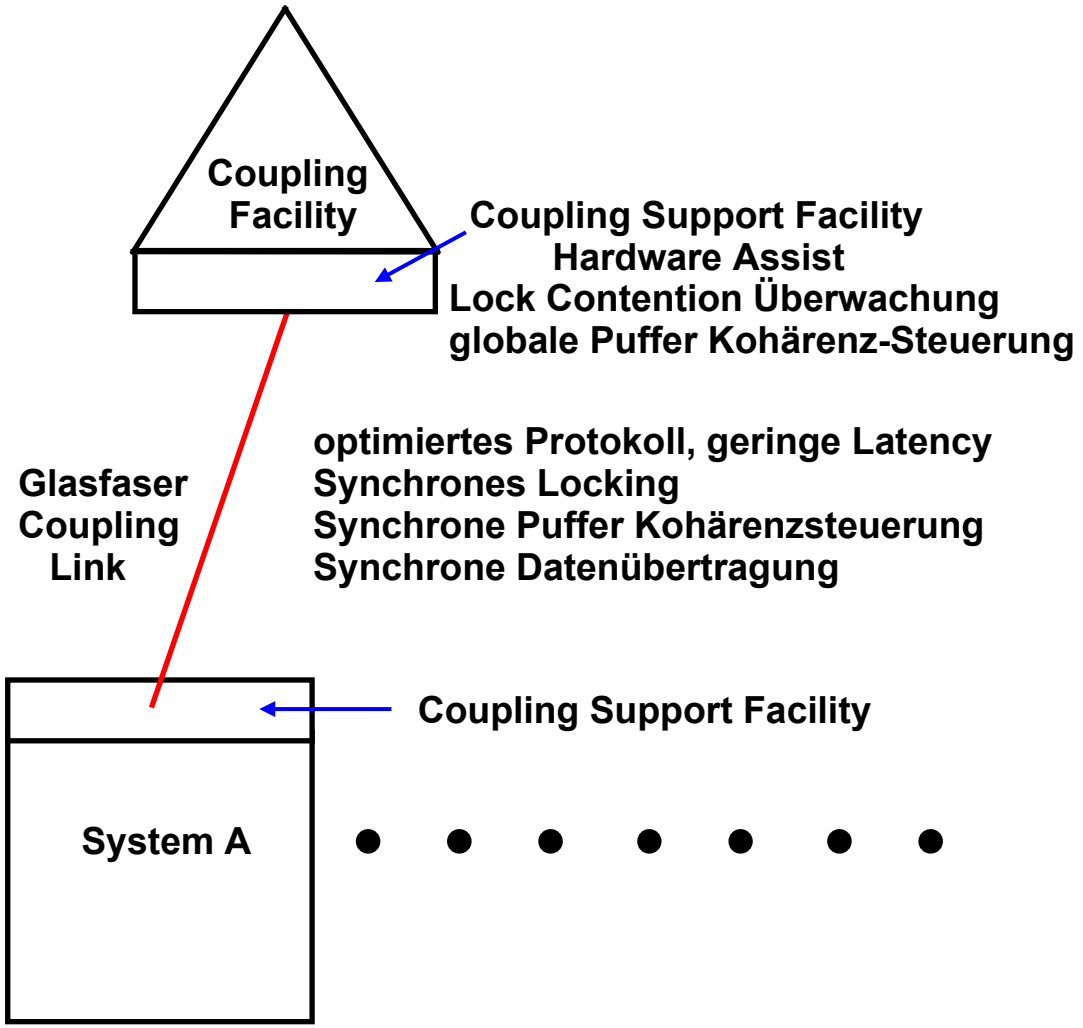
Coupling Support Facility

Jeder Knoten (System) eines Sysplex sowie die CF selbst enthalten eine spezielle Hardware Einrichtung, die **Coupling Support Facility**. Diese bewerkstelligt die Kommunikation und den Datenaustausch zwischen Knoten und CF.

Die **Coupling Support Facility** besteht aus je einer Glasfaserverbindung (800 MByte/s theoretisches Maximum), einem dedizierten Link-Prozessor für die Verarbeitung des (Infiniband) Übertragungsprotokolls und einer Erweiterung des System z Maschinenbefehlssatzes. Die zusätzlichen Maschinenbefehle realisieren die nahe Kopplung zwischen Knoten und CF und ermöglichen allen Knoten den Zugriff auf die Inhalte in den CF-Datenstrukturen.

Das Kommunikationsprotokoll wurde für eine besonders geringe Latenzzeit optimiert, welches *synchrone Zugriffe* eines jeden Knotens auf die CF ermöglicht. Synchron bedeutet, dass ein auf einem Knoten laufender Prozess während eines Zugriffs auf die Coupling Facility blockiert: Er bleibt im Zustand „running“, und es findet kein Prozesswechsel und kein Übergang "User-Status - Kernel-Status" statt. Die Zugriffsgeschwindigkeit liegt im Mikrosekundenbereich und ist somit weitaus effizienter als die Kommunikation über allgemeine Protokolle bei loser Rechnerkopplung.

Anbindung eines Systems an die Coupling Facility



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 (Infiniband) mit besonders geringer Latency eingesesetzt .

Die CF Glasfaser Verbindung wird durch spezielle Hardware Einrichtungen und durch zusätzliche Maschinenbefehle in jedem angeschlossenen System unterstützt (Coupling Support Facility).

Die Coupling Facility kann in einem angeschlossenen Rechner ohne Unterbrechung des laufenden Prozesses Daten in spezielle Speicherbereichen (Bit Vektoren) abändern.

Komponenten einer Coupling Facility

Die nahe Kopplung über die CF wird für die leistungskritischen Kontrollaufgaben in Shared-Disk-Clustern genutzt, um durch die effiziente Realisierung eine hohe Skalierbarkeit zu erreichen. Dies betrifft die globale Synchronisation über Sperrverfahren (Locking), die Kohärenzkontrolle der Pufferinhalte mit schnellem Austausch geänderter Daten zwischen den Knoten sowie die flexible Lastverteilung. Entsprechend werden spezifische Funktionen (Maschinenbefehle) für Locking, Caching und Queuing sowie zugeschnittene Datenstrukturen im Hauptspeicher der CF bereitgestellt. Die Hauptspeicher-Ressourcen der CF-können dynamisch partitioniert und einer der CF-Strukturen zugewiesen werden. Innerhalb derselben CF sind mehrere CF-Strukturen desselben oder unterschiedlichen Typs möglich.

Die auf diesen Strukturen bereitgestellten Funktionen bzw. Cluster-Protokolle repräsentieren drei Verhaltensmodelle:

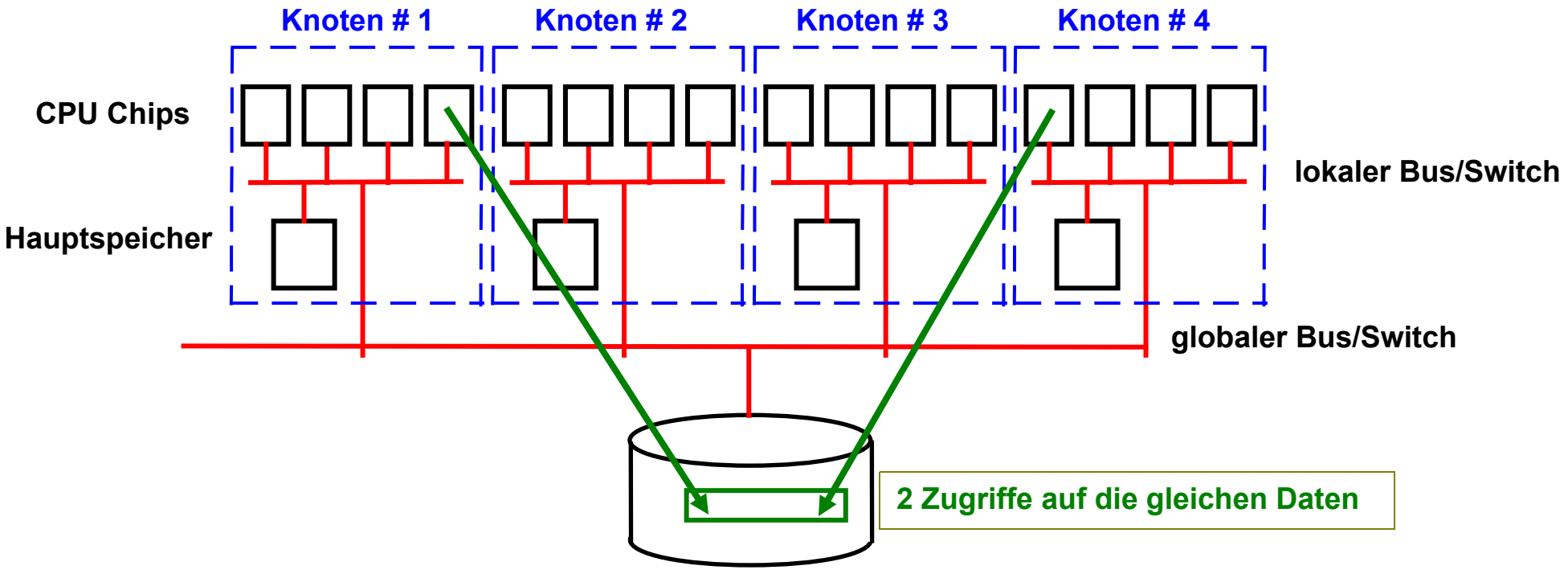
Host Hardware: Wenn eine CPU eine Anforderung an die Coupling Facility startet, wird diese zunächst mittels Firmware und spezieller Hardware verarbeitet, ehe sie über das Coupling Link weitergereicht wird.

Coupling Link: Die Anforderung wird dann über das Coupling Link an die Coupling Facility übertragen. Die Übertragungszeit wird durch die Link Geschwindigkeit, die Datenmenge, die Distanz zur Coupling Facility (Lichtgeschwindigkeit) bestimmt. Die Übertragung erfolgt nach Möglichkeit synchron, d.h. der laufende Prozess wird nicht unterbrochen. Deshalb ist die Latenz (die Zeit, die für die Übertragung eines einzelnen Bytes benötigt wird) besonders kritisch. Um die Latenzy zu minimisieren, verwenden CF Links deshalb entweder ein spezielles CF Link Protokoll, oder das Infiniband Protokoll. Beide zeichnen sich durch eine besonders geringe Latency aus.

Coupling Facility Hardware: Die eintreffende Anforderung wird durch die Hardware und den Code in der Coupling Facility (Coupling Facility Control Code, CFCC) bearbeitet. Je nach Bedarf sendet die Coupling Facility Nachrichten an die Knoten Rechner des Sysplex zurück.

Gleichzeitiger Zugriff zu Daten im Cluster

Cluster haben den Vorteil, dass mehrfache Instanzen (Kopien) des Betriebssystems vorhanden sind. Deshalb bestehen alle modernen Großrechner aus einem Cluster von SMPs.



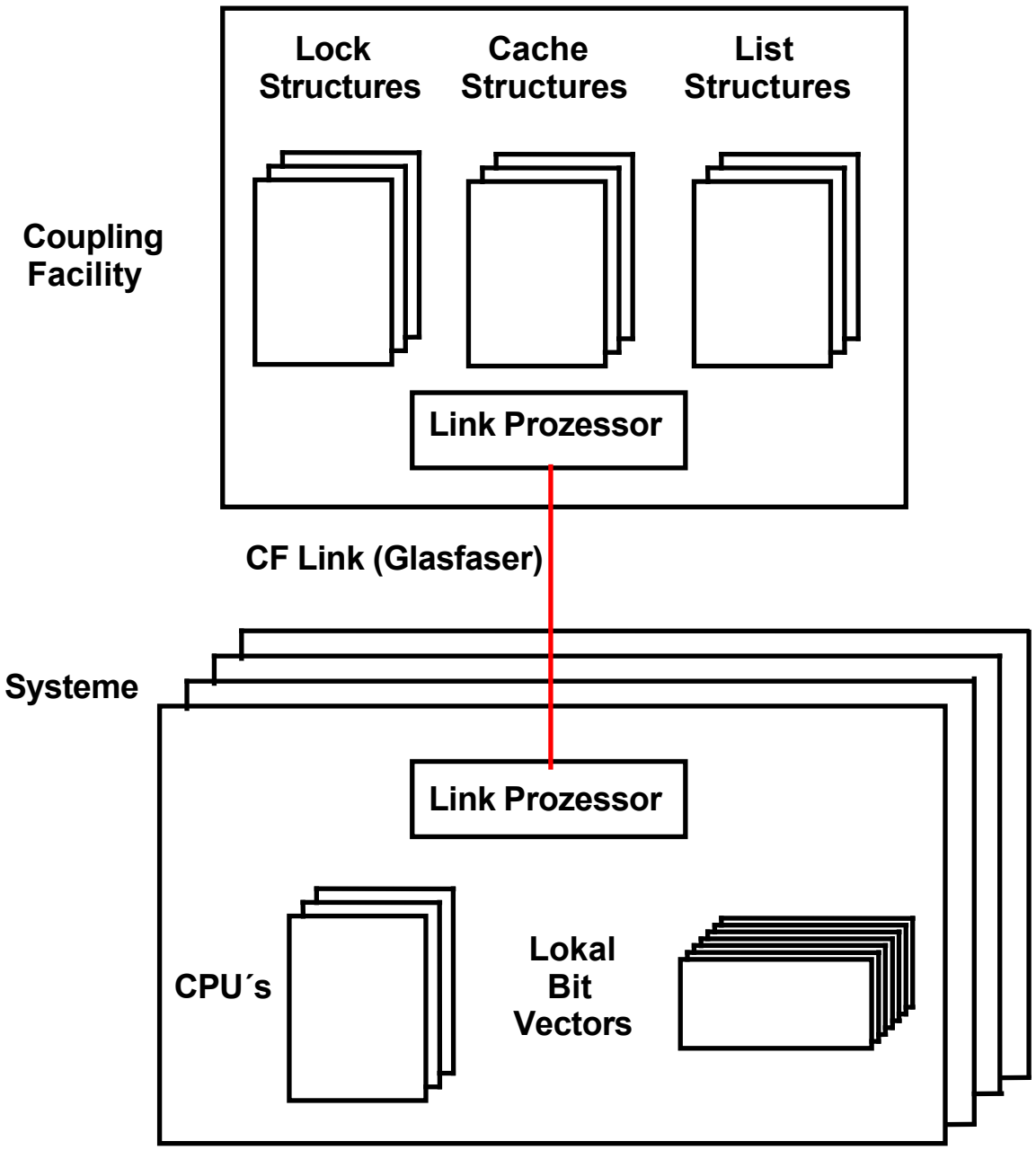
Allerdings besteht hierbei das Problem, den gleichzeitigen Zugriff mehrerer Knoten auf den gleichen Datenbestand zu lösen. Dies geschieht mit Hilfe von Locks (Sperren) auf Teile der gemeinsam genutzten Daten.
Die Effektivität des Lock Managements (Sperrverwaltung) bestimmt die Skalierungseigenschaften eines Clusters.

Coupling Facility Strukturen

Die nahe Kopplung über die CF wird für die leistungskritischen Steuerungslaufgaben genutzt, um durch die effiziente Realisierung eine hohe Skalierbarkeit zu erreichen. Dies betrifft die globale Synchronisation über Sperrverfahren (Locking), die Kohärenzkontrolle der Pufferinhalte mit schnellem Austausch geänderter Daten zwischen den Knoten sowie die flexible Lastverteilung. Entsprechend werden spezifische Funktionen (Maschinenbefehle) für Locking, Caching und Queuing sowie zugeschnittene Datenstrukturen im Hauptspeicher der CF bereitgestellt. Die CF Hauptspeicher-Ressourcen können dynamisch partitioniert und einer der CF-Strukturen zugewiesen werden. Innerhalb derselben CF sind mehrere CF-Strukturen desselben oder unterschiedlichen Typs möglich.

Die auf diesen Strukturen bereitgestellten Funktionen bzw. Cluster-Protokolle repräsentieren drei Verhaltensmodelle:

- **Lock-Modell**: Es unterstützt feingranulares, globales Locking für hohe Transaktionsverarbeitungs-Performance und eine Signalisierung von Zugriffskonflikten.
- **Cache-Modell**: Es liefert eine globale Kohärenz-Steuerung für die verteilten Pufferinhalte der einzelnen Knoten sowie einen globalen Puffer in der CF (Shared Data Cache).
- **List Modell (Queue-Modell)**: Es implementiert einen umfangreichen Satz an Listen und Queuing-Konstrukten für die Verteilung von Arbeitslasten, zur Realisierung einer schnellen Nachrichtenübertragung (Message Passing) sowie für die Verwaltung globaler Status- Informationen. Ein Beispiel für letzteres ist die Verwaltung globaler Sicherheitsrechte.



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 (Teil der Coupling Support Facility) direkt verbunden. Die Kommunikation CPU – CF wird durch spezifische Maschinen-befehle unterstützt.

Jedes System unterhält lokale, als „Bit Vectoren“ bezeichnete Speicher-bereiche für eine logische Verbindung zu jeder Struktur in der CF. Es existieren getrennte Bit Vektoren für die Lock Strukturen, Cache Strukturen und List (Queue) Strukturen.

Zuordnung von Bit Vektoren zu CF Strukturen

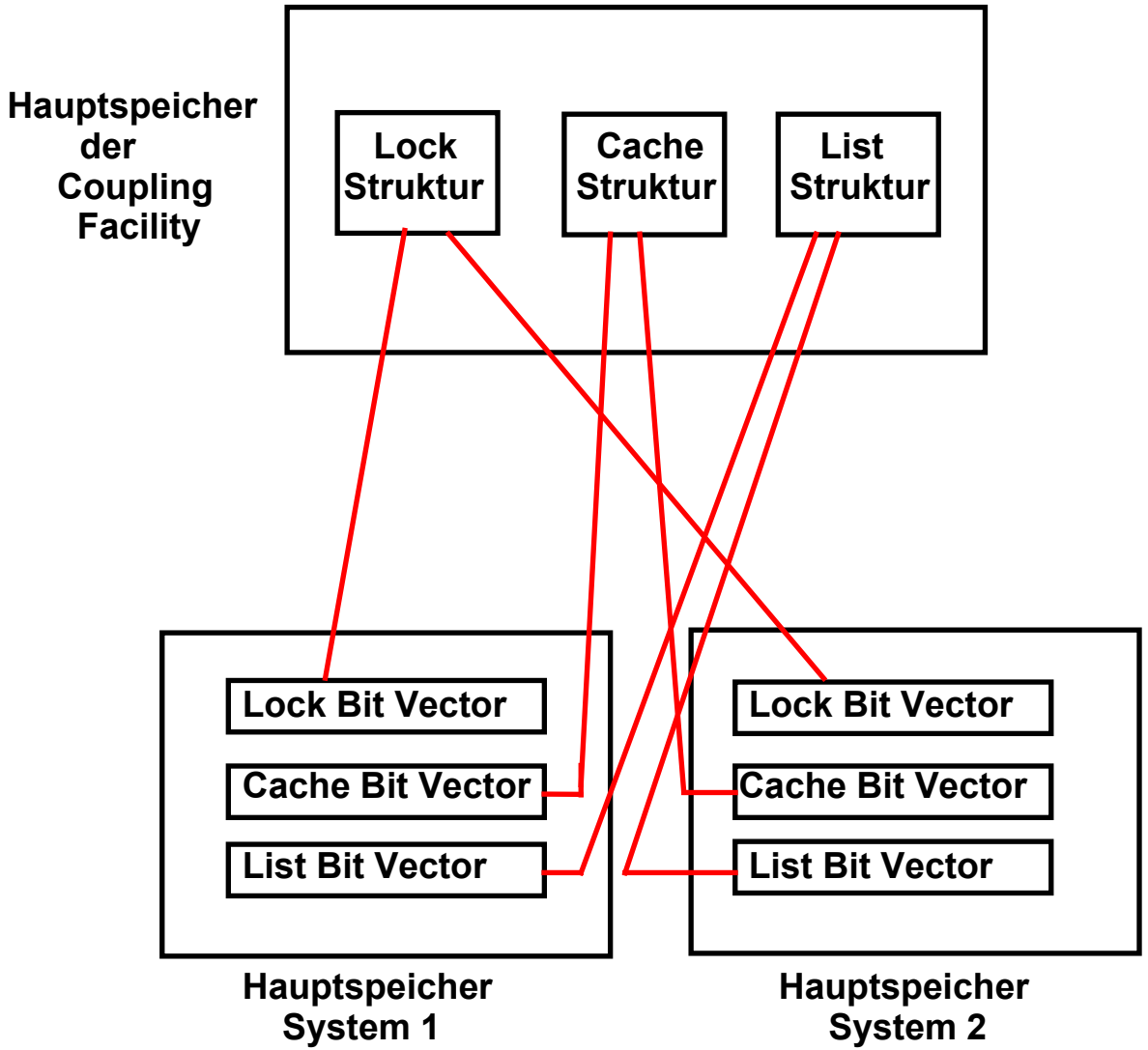
Die folgende Abbildung zeigt die Anbindung der Knoten (Systeme) an die CF und die dort verwalteten globalen Datenstrukturen.

Jeder angeschlossene Rechner dupliziert in seinem Hauptspeicher eine Untermenge der in der Coupling Facility gespeicherten Daten. Diese Untermengen werden als Bit Vektoren bezeichnet.

Der Zugriff erfolgt dabei nicht nur von den Systemen (Knoten) auf die CF, sondern die CF kann auch direkt (ohne Involvierung des Betriebssystems) auf bestimmte Hauptspeicherinhalte der Systeme, spezifisch die Bit-Vektoren, zugreifen und ändern. Solche Bit-Vektoren existieren für jede logische Verbindung zu einer CF-Datenstruktur und erlauben z.B. die schnelle Signalisierung von Zugriffskonflikten (s.u.).

In den nächsten Abschnitt betrachten wir die Nutzung der CF-Lock- und Cache-Strukturen zur Realisierung der clusterweiten Synchronisation (Locking) und Kohärenzkontrolle.

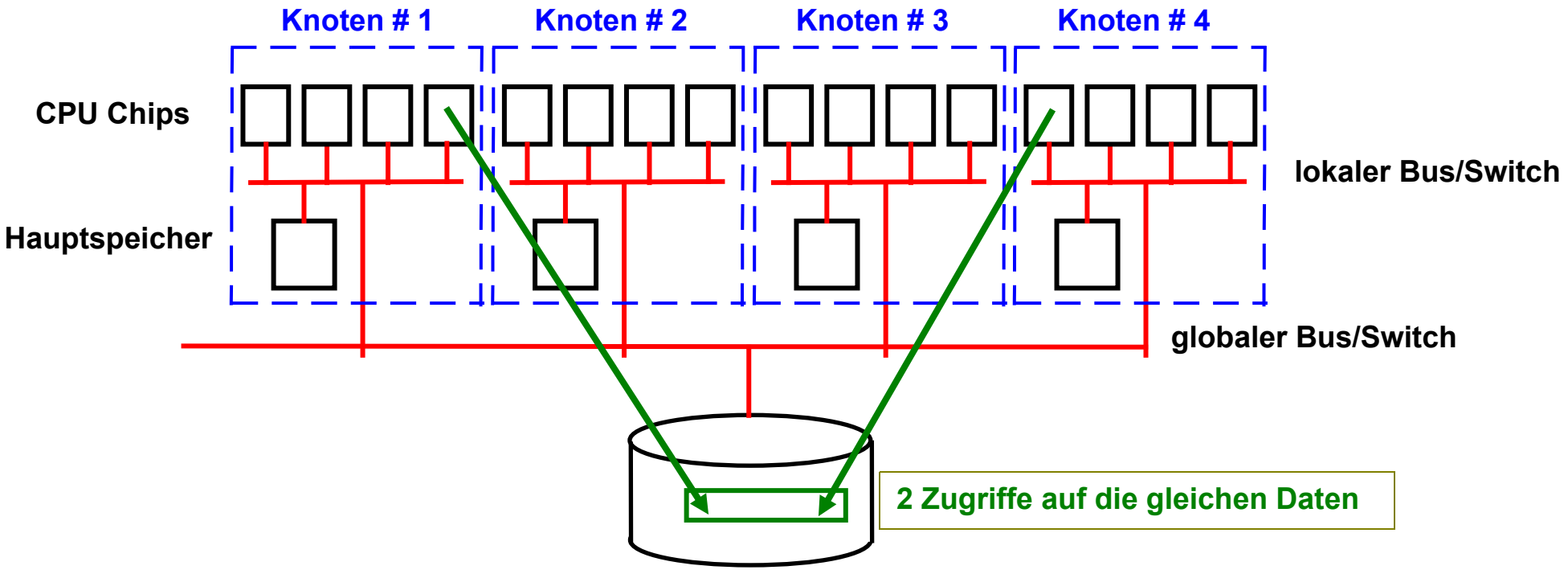
Zuordnung von Bit Vektoren zu CF Strukturen



Eine Untermenge der in der Coupling Facility gespeicherten Information ist in den Hauptspeichern der beteiligten Systeme in der Form von Bit Vektoren dupliziert.

Gleichzeitiger Zugriff zu Daten im Cluster

Cluster haben den Vorteil, dass mehrfache Instanzen (Kopien) des Betriebssystems vorhanden sind. Deshalb bestehen alle modernen Großrechner aus einem Cluster von SMPs.



Allerdings besteht hierbei das Problem, den gleichzeitigen Zugriff mehrerer Knoten auf den gleichen Datenbestand zu lösen. Dies geschieht mit Hilfe von Locks (Sperren) auf Teile der gemeinsam genutzten Daten.
Die Effektivität des Lock Managements (Sperrverwaltung) bestimmt die Skalierungseigenschaften eines Clusters.

Locking Problem

Angenommen zwei Transaktionen, die auf unterschiedlichen Systemen des Sysplex laufen.

Beispiel: Beide Transaktionen greifen auf die zwei Variablen d1 und d2 zu.

Anfangswerte: d1 = 15, d2 = 20 .

Transaktion 1	Transaktion 2

read d1
if d1 > 10
●
●
sub d1, 10
●
add d2, 10
●
end if
	read d1
	if d1 > 10
	●
	sub d1, 10
	●
	add d2, 10
	●
	end if

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.

Concurrency Control

- **Concurrency Control (Synchronisierung)** ist zwischen den beiden Transaktionen erforderlich, um dies zu verhindern.
- Erforderlich für Application Server, die Daten selbst speichern und auch auf Datenbanken zugreifen.
- Der Begriff, der dies beschreibt, ist **Serialisierung (serializability)**
- Es gibt viele Implementierungsmöglichkeiten.
  Gebräuchlich ist „**two-phase locking**“ (nicht zu verwechseln mit dem two-phase Commit Protokoll)

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 über 4 Methoden und zwei Zustände S und E verfügt.

Die Methode

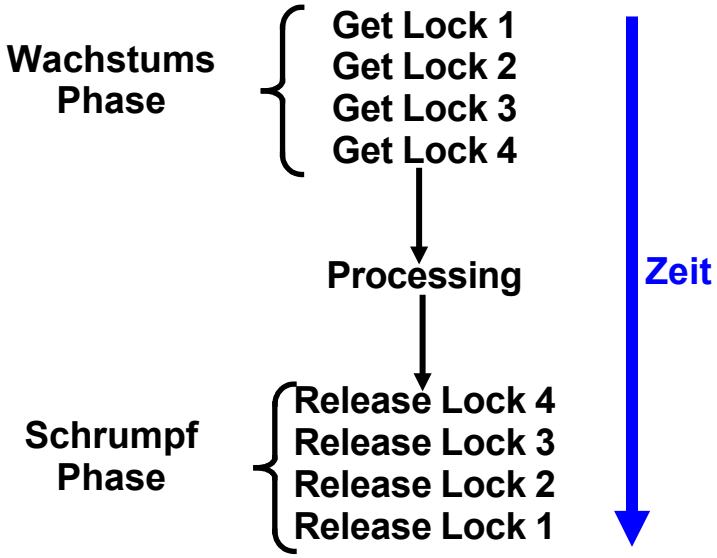
- **GetReadLock** reserviert S Lock (shared),
- **GetWriteLock** reserviert E Lock (exclusive),
- **PromoteReadtoWrite** bewirkt Zustandswechsel S → E,
- **Unlock** gibt Lock frei.

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 Locking 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.

2-Phase Locking



Es werden alle Locks gesetzt, ehe die Verarbeitung beginnt. Wenn das nicht möglich ist, werden **alle** Locks sofort wieder freigegeben, und ein neuer Versuch beginnt. Nach Abschluss der Verarbeitung werden die Locks wieder frei gegeben.

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 Shared	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 S Lock für ungültig erklärt.

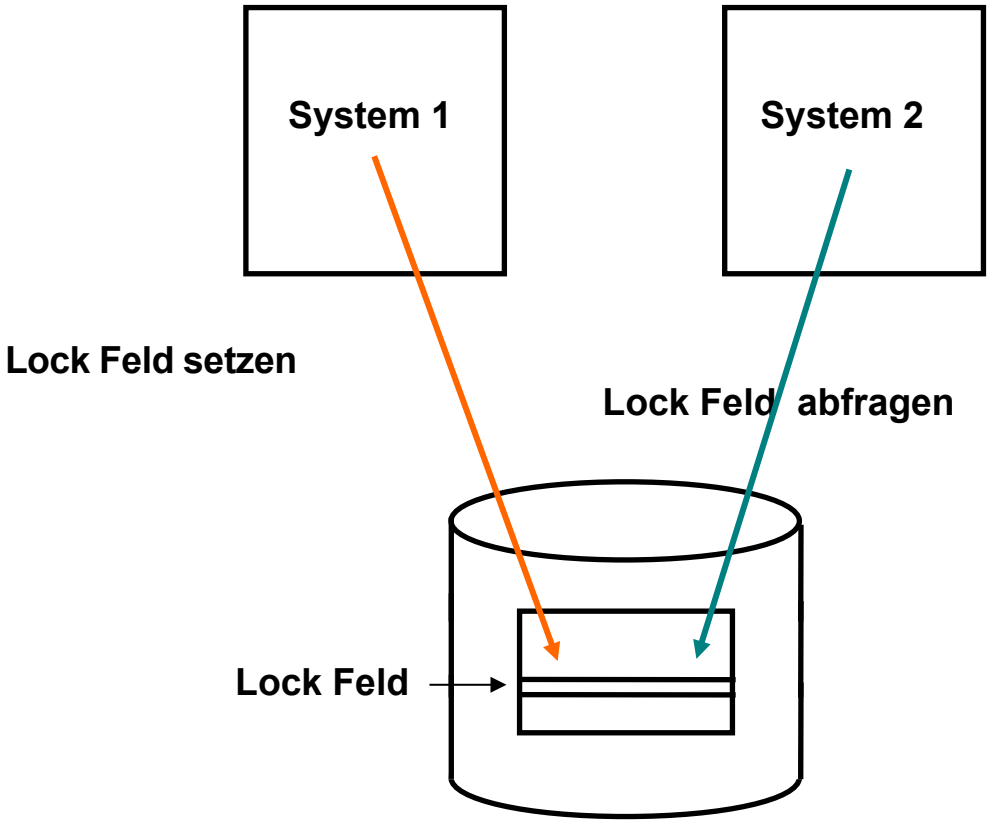
Das Senden einer Nachricht informiert Transaktion 2, dass ihr Wissen über den Zustand der beiden Variablen d1 und d2 nicht mehr gültig ist.

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) ?

Sysplex Teil 3

Lock-Strukturen

Lock-Verwaltung



Frage: Wie speichert man die Locks?

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

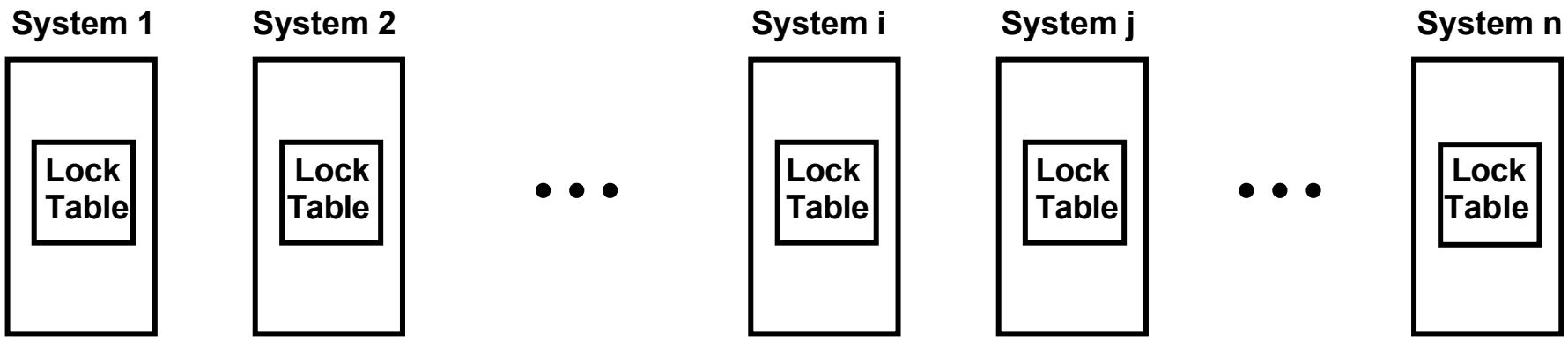
Jeder zu schützende Datenbereich auf der Festplatte (z. B. eine Zeile in einer relationalen Datenbank-Tabelle) erhält ein zusätzliches Lock-Feld.

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.

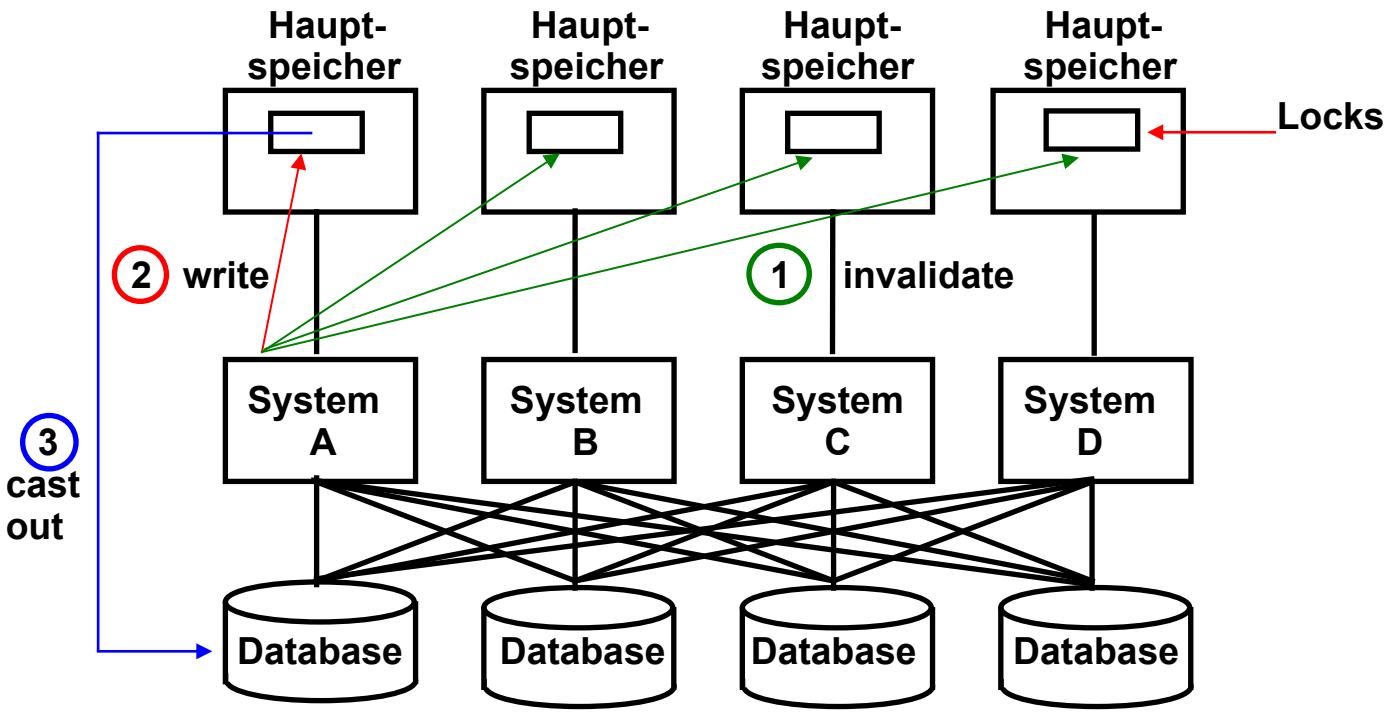
Bei einem Cluster mit mehreren SMPs und mehreren Hauptspeichern sind auch mehrere Lock Tabellen in den Hauptspeichern der beteiligten Systeme vorhanden, die alle auf dem gleichen Stand gehalten werden müssen (verteilte Lock-Tabelle).

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.

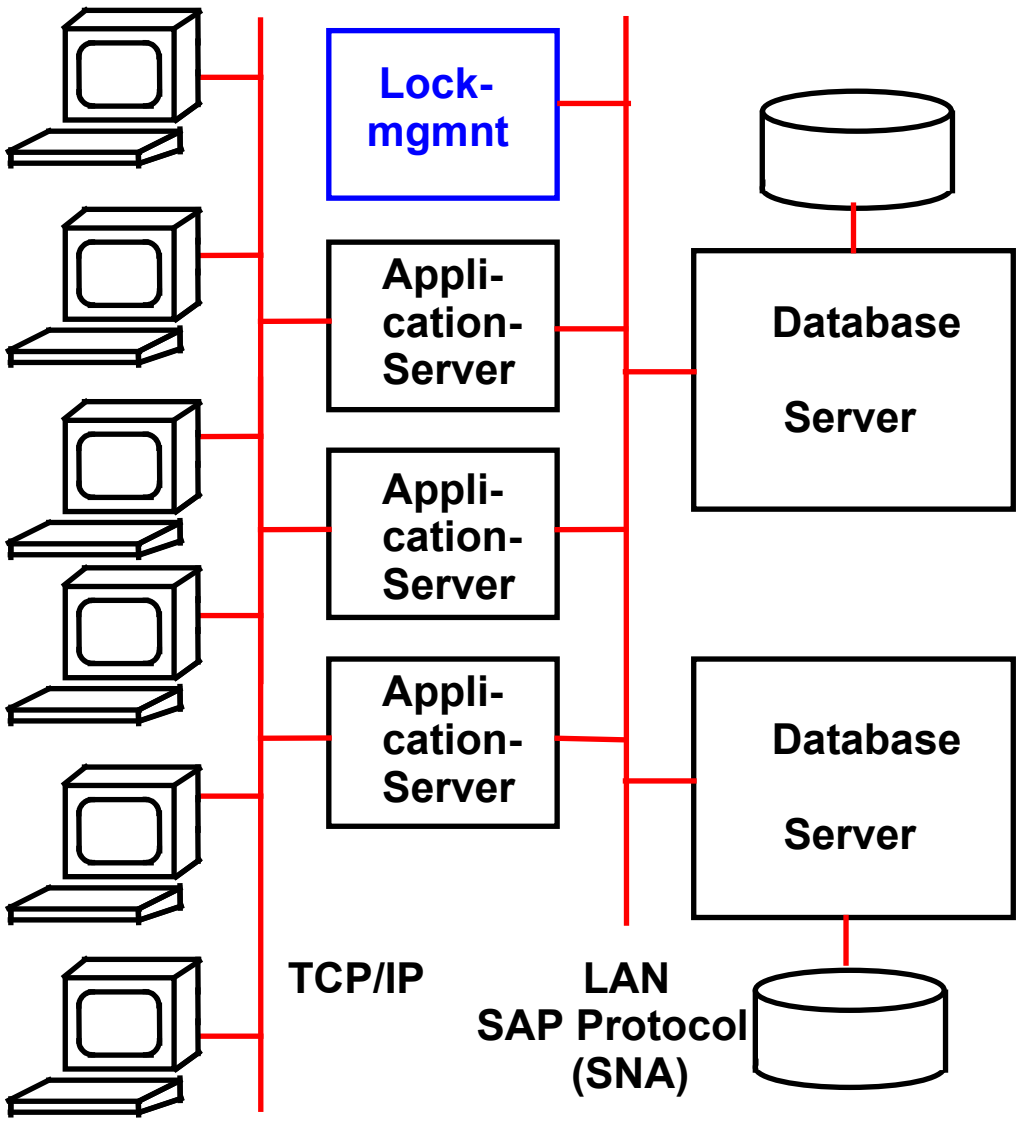
Beispiele für diese Lösung sind die VAX DBMS und VAX Rdb/VMS Datenbanksysteme. (Gray/Reuter p.380)

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, Transaction Processing - Concepts and Techniques 1993)



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.

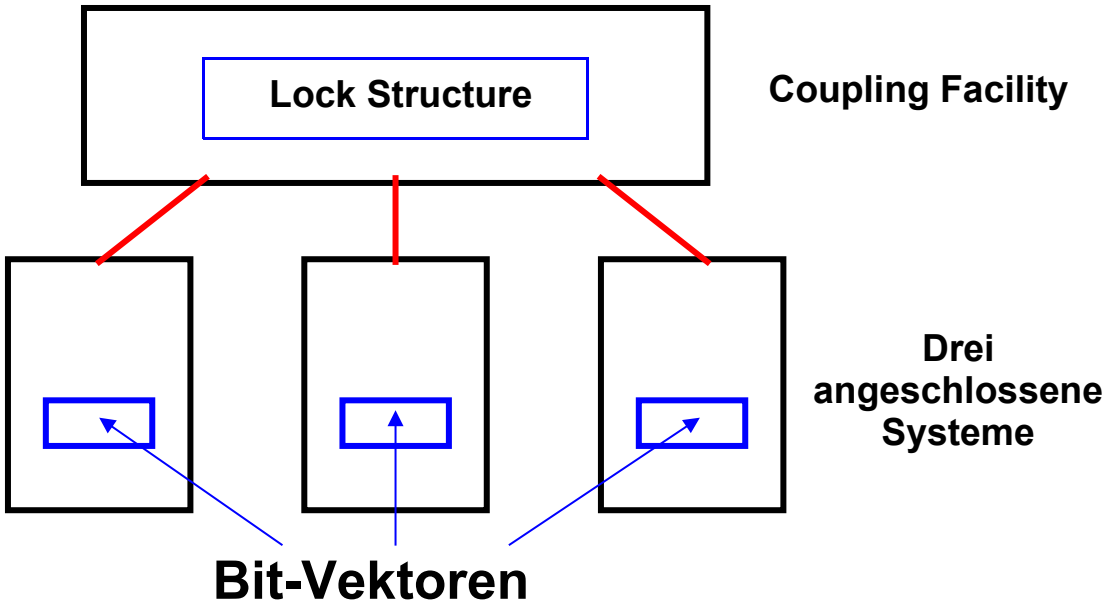
Eine typische SAP/R3 Konfiguration verwendet einen getrennten Server für das Lock Management (Sperr-Server). Wenn ein Knoten ein S Lock erwerben möchte, wendet er sich an den Sperr-Server um zu erfragen, ob ein anderer Knoten ein Interesse an diesem Lock hat. Wenn ja, wird nur dieser benachrichtigt.

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.

Typische SAP/R3 configuration

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 Eintrag 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 Kenntnisnahme 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.

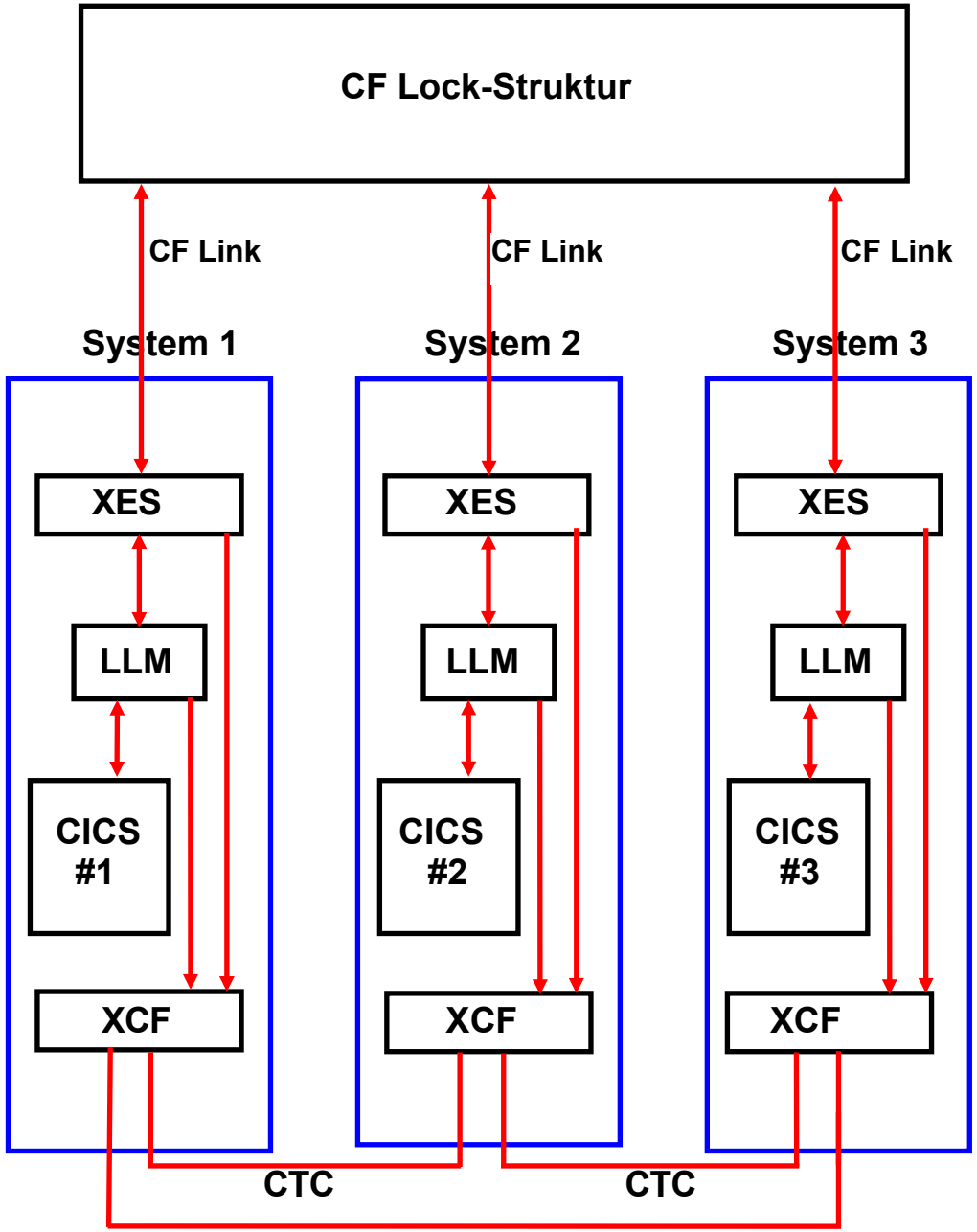
In der CF-Lock-Struktur wird eine zentrale Locktabelle verwaltet, auf die LLMs aller Knoten zugreifen, um eine schnelle systemweite Synchronisation konkurrierender Datenzugriffe zu erreichen. Diese Zugriffe erfolgen synchron über die Coupling Links und eine als „Cross System Extension Services" (XES) 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 der 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 Cross System Extension Services (XES) 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 sein.

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?

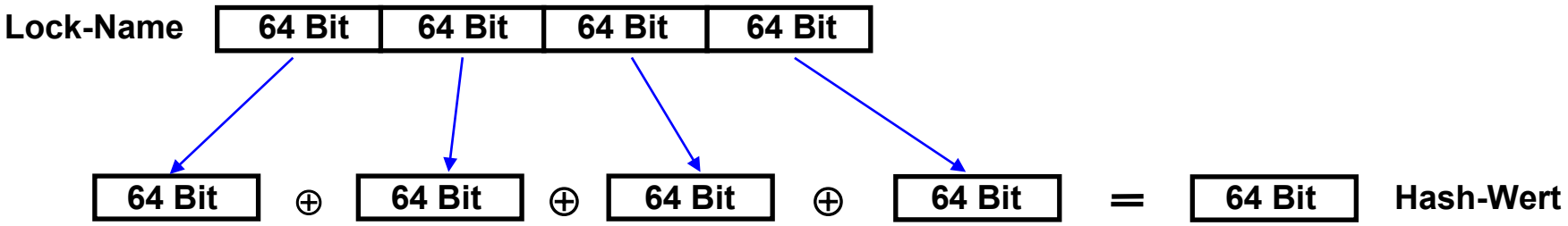
Das Problem ist, dass Lock-Namen sehr lang sein können. Zum Beispiel haben Locks in IMS-Datenbanken bis zu 19 Bytes (152 Bits) lange Namen. Da eine Locktabelle mit 2^{152} Einträgen für den Hauptspeicher viel zu groß (und sehr dünn belegt) wäre, erfolgt eine Abbildung in eine Hash-Klassen-Bezeichnung 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



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 (Außerdem muss der Hash-Algorithmus performant in Hardware und/oder Software – je nach Einsatzzweck - implementierbar sein).

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.

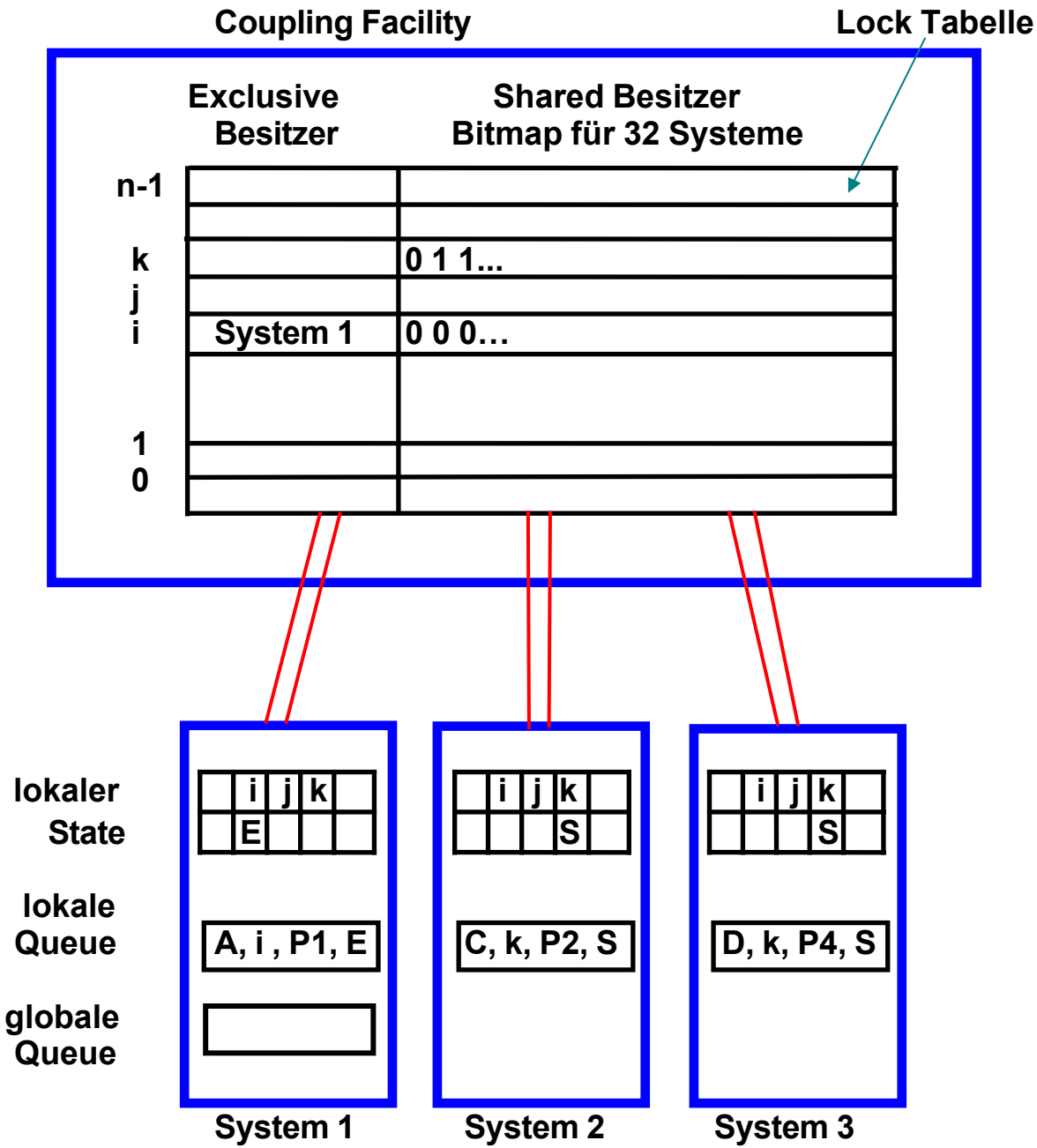
Wenn die Lock-Tabelle 20.000.000 Einträge hat, beträgt die Wahrscheinlichkeit etwa 0,5 %, dass 2 Locks zufällig in der gleichen Hash-Klasse abgebildet werden. 16 777 216 Einträge erfordern eine Hasch Wert mit einer Länge von 24 Bit.

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 Lese-Locks (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 ein 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) ist 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.

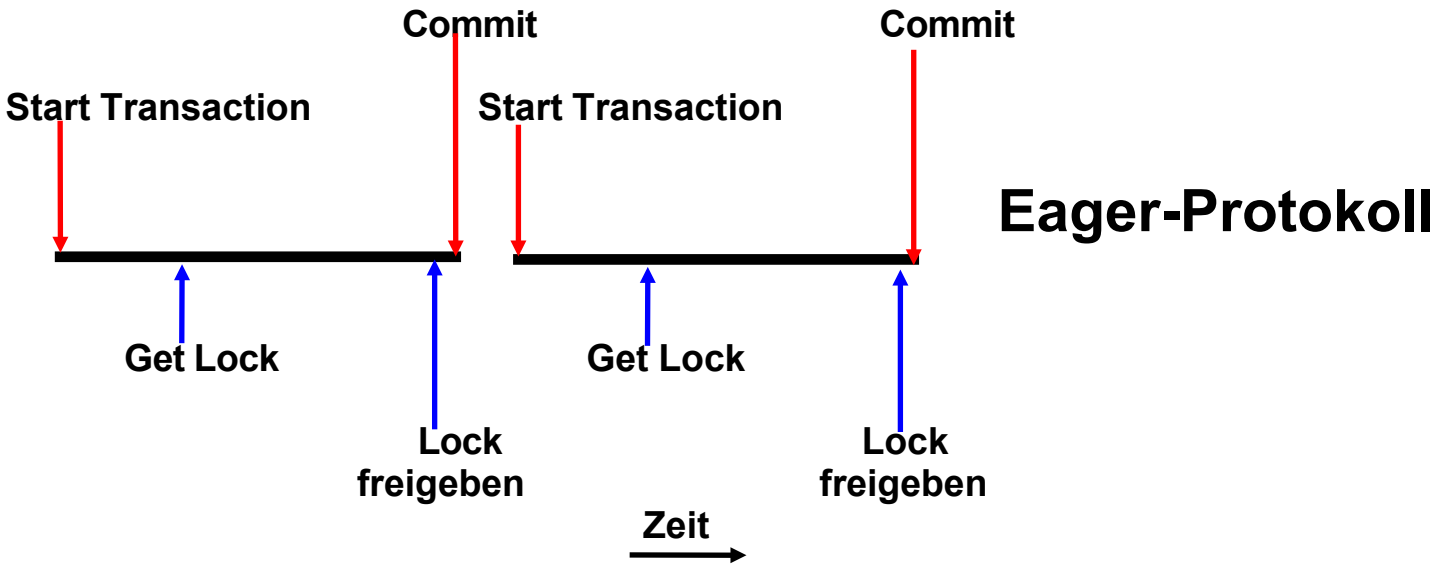
   In der lokalen Queue von System 1 wird festgehalten, dass Lock A, Hash-Klasse i von dem lokalen Prozess P1 mit der Berechtigung Exclusive gehalten wird.

   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 Bitmap 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.

Locking-Protokolle



Eager-Protokoll Lock freigeben, wenn Commit die Transaction beendet (siehe das obige Beispiel).

Lazy-Protokoll Lock freigeben, wenn eine Contention auftritt; nichts tun, bis jemand anderes das Lock benötigt.

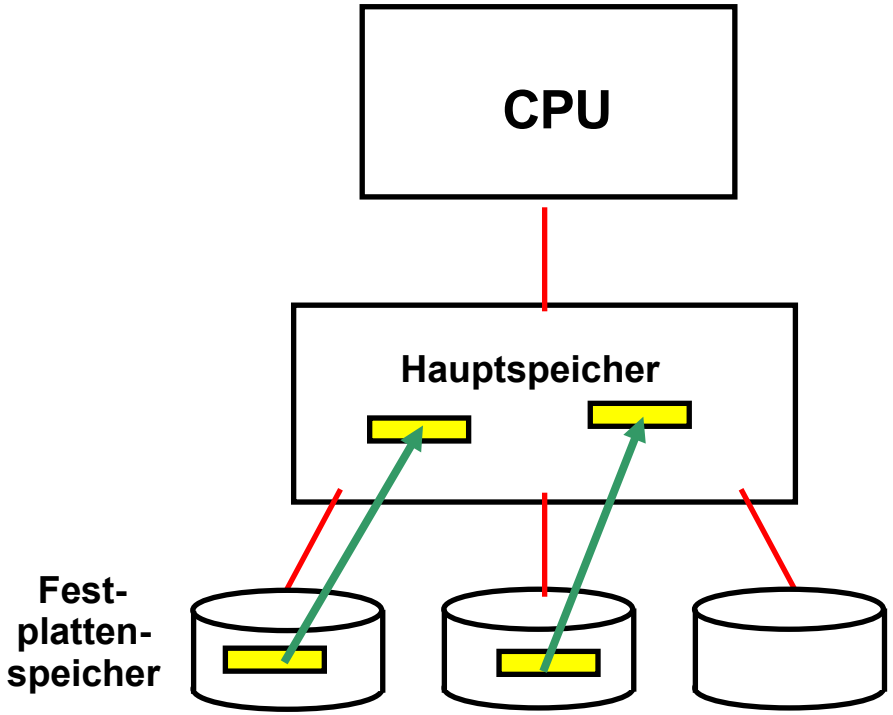
Das Lazy-Protokoll arbeitet besser, wenn Datenkonflikte selten auftreten. Ein Beispiel ist das TPC-C Benchmark des Transaction Processing Councils (www.tpc.org). Hersteller von Datenbank-Software benutzen TPC-C gerne, weil dann ihr Produkt besser aussieht. Die Unterschiede zu praktischen Anwendungen können allerdings sehr groß sein.

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

Sysplex Teil 4

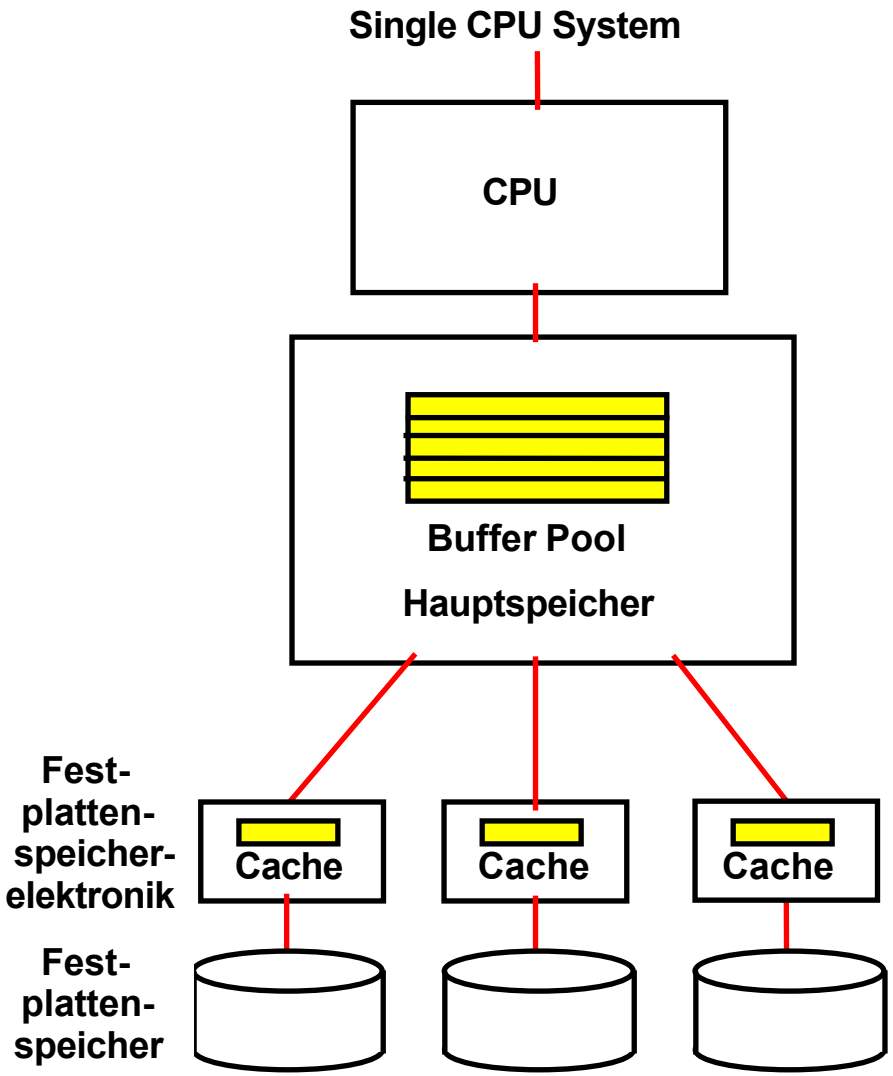
Cache- und Listen Strukturen

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 Datenbank-System optimal verwaltet.



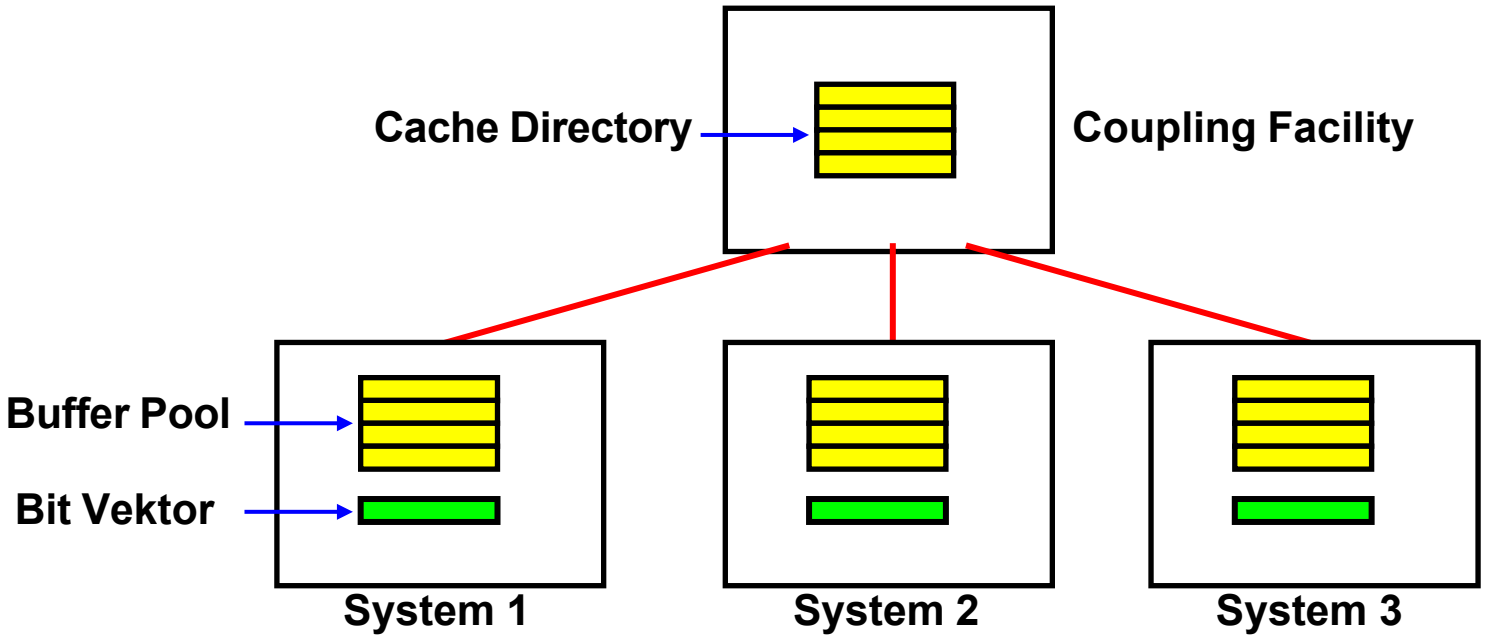
Festplattenspeicher Cache und Hauptspeicher Buffer Pool

Der Buffer Pool stellt eine Art Plattenspeicher Cache im Hauptspeicher dar. Das Datenbank-System bemüht sich, den Speicherplatz im Buffer Pool optimal zu verwalten. (Unabhängig davon werden Daten zusätzlich in einem Plattenspeicher Cache gespeichert, der Bestandteil der Plattenspeicher Elektronik oder des Enterprise Storage Servers ist).

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

In einem Cluster ist nicht auszuschließen, dass Datensätze oder Datenbank Records 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 Eintrag für jeden Buffer in den angeschlossenen Systemen befindet. Analog zur Lock-Verwaltung befinden sich außerdem in jedem System Bit-Vektoren, die den Inhalt des Cache Directory 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

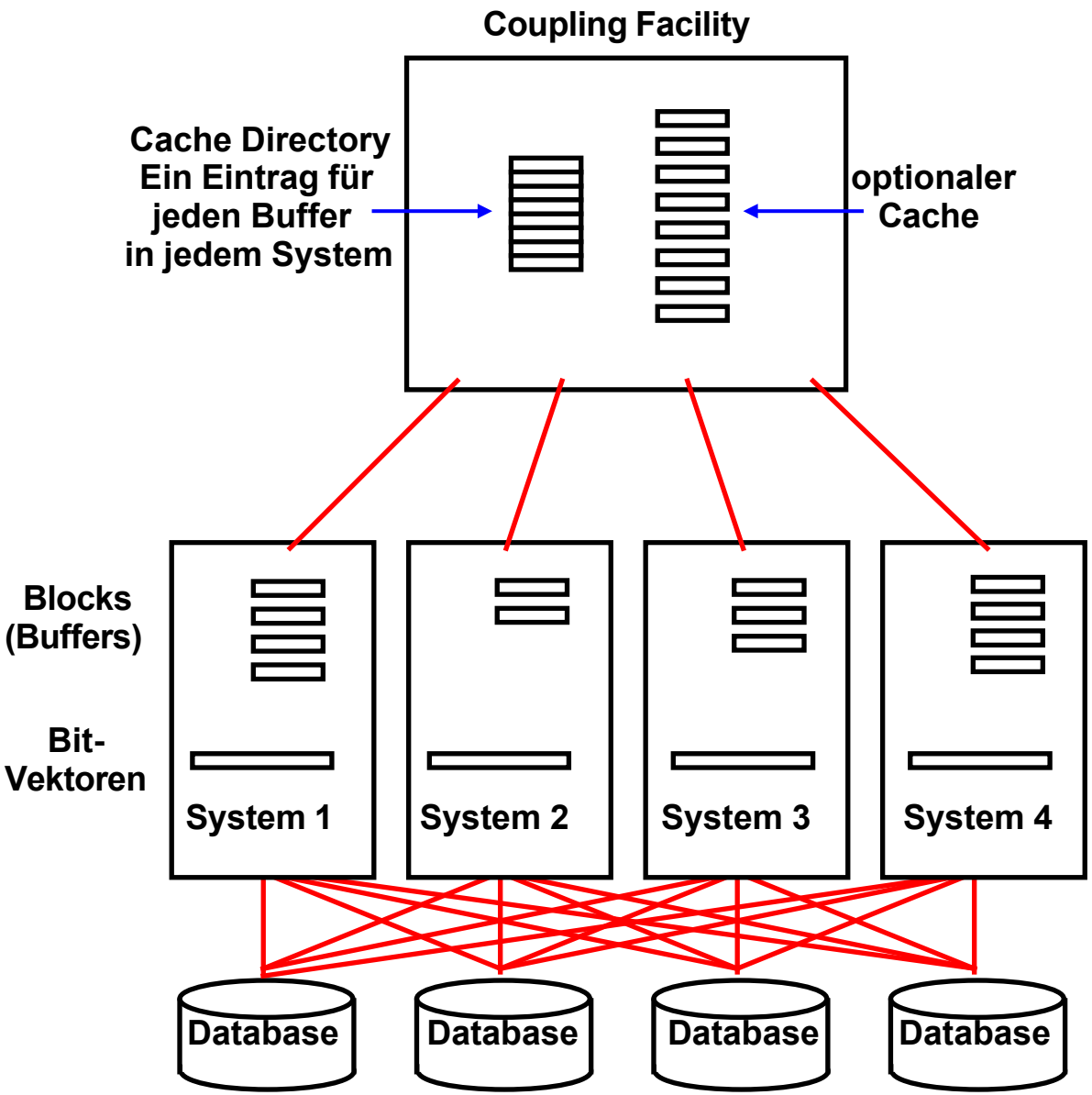
Der lokale Buffer Pool im System 1 enthält Puffer (Blöcke) mit Records, die gerade bearbeitet werden. Solange die Transaktion nicht abgeschlossen ist, verhindert der Lock-Manager einen Zugriff durch ein anderes System (z.B. System 2).

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

Greift System 2 jetzt auf einen Buffer mit dem gleichen Datenbankrecord 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 Directory durch System 1.
Die CF sendet hierzu eine „Cross-Invalidate“ (CI) Nachricht an alle anderen betroffenen Systeme (und nur an die betroffenen Systeme)

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



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.

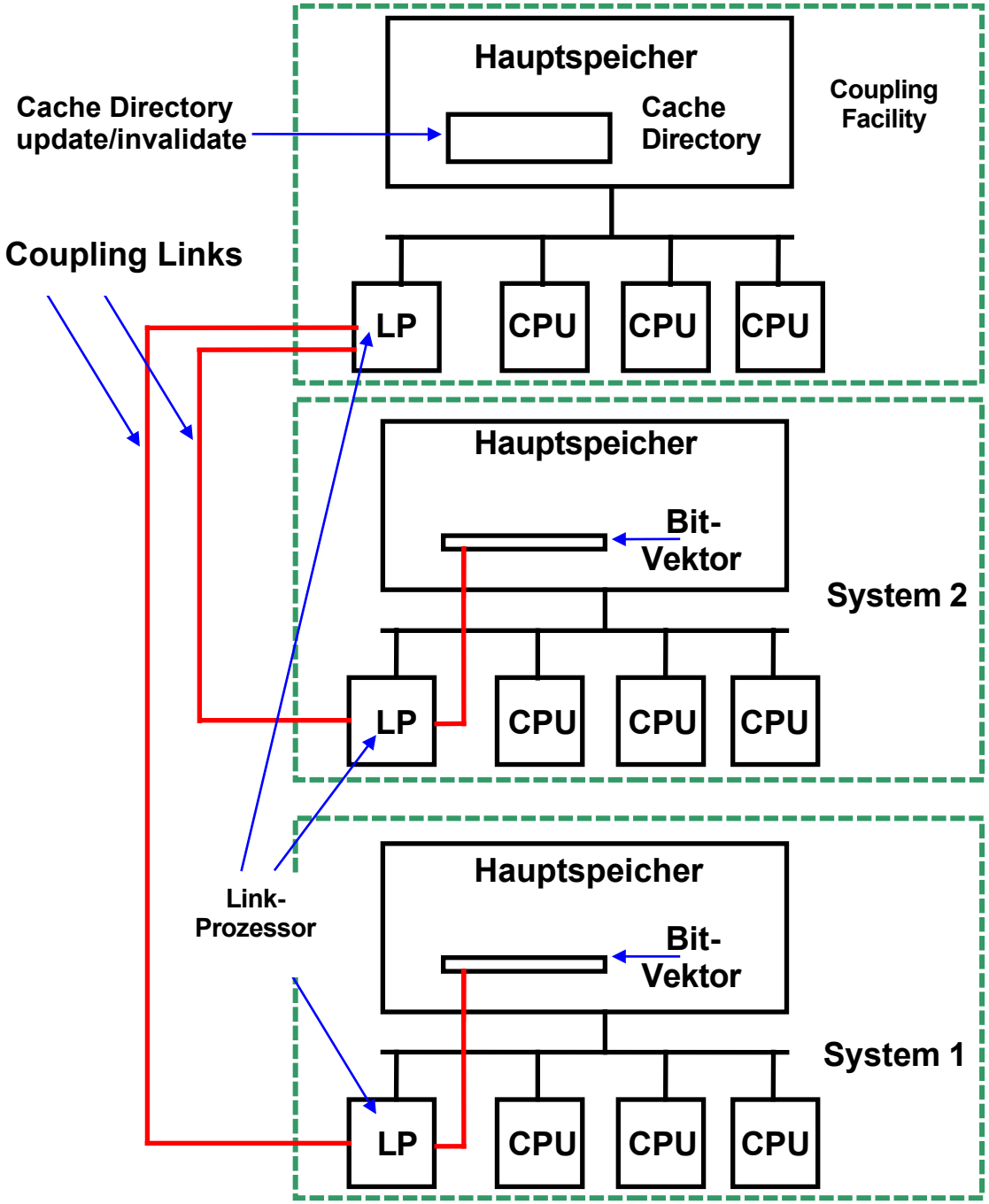
a) System 1 Read from Disk
1. Load Block from Disk
2. Register with CF Directory
3. add Bit in Bit Vector

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. detects invalid Bit im localen Bit Vector
3. führt erforderliche Maßnahmen durch



Beim Force-at-Commit sendet die CF eine „Cross-Invalidate (CI) Nachricht an alle anderen Systeme.

Link-Prozessoren (LP) haben einen Direct Memory Access (DMA) zu dem Hauptspeicher.

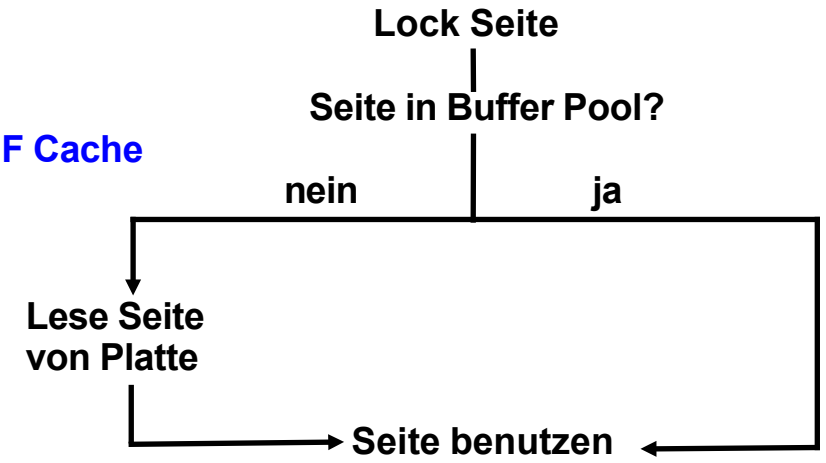
Über die Link-Prozessoren der Coupling Facility und die Link-Prozessoren der Systeme können Bit-Vektoren im Hauptspeicher abgeändert werden, ohne dass der normale Programmablauf in den Systemen dadurch beeinflusst wird (kein Prozesswechsel). Dies verbessert das Leistungsverhalten, da jeder Prozesswechsel eine Pfadlänge von mehreren Tausend Maschinenbefehlen erfordert.

Coupling Facility Cache

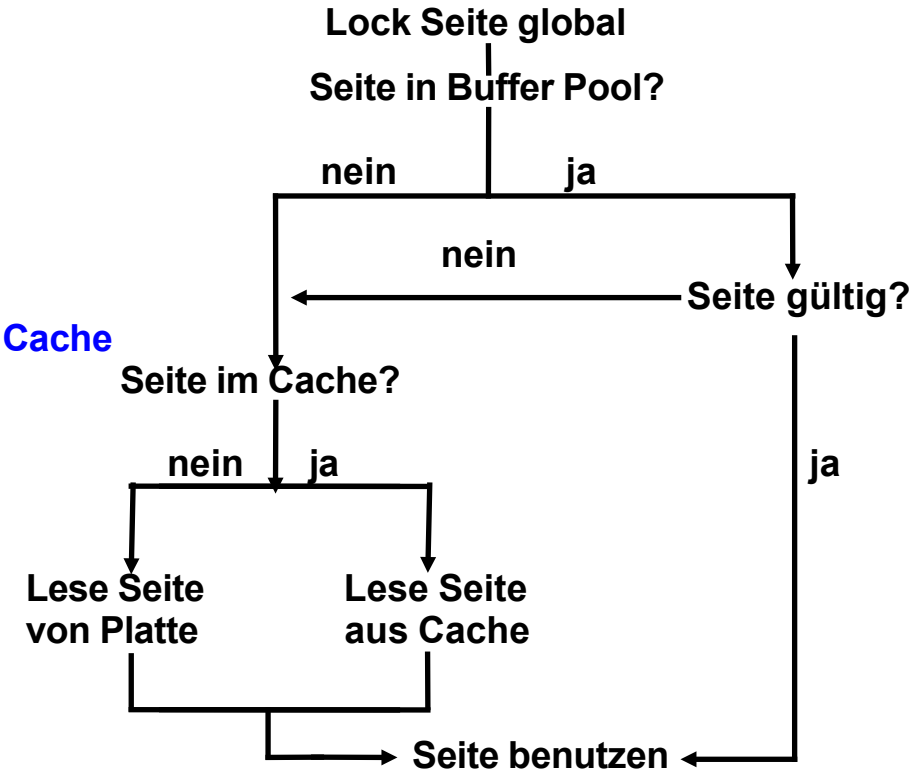
Neben dem Cache Directory kann die Coupling Facility auch als Plattenspeicher-Cache genutzt werden. Hiervon machen einige, aber nicht alle Datenbanksysteme, Gebrauch. DB2 und Adabas nutzen die Coupling Facility auch als Plattenspeicher-Cache, IMS jedoch nicht.

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

a) ohne CF Cache



b) mit CF Cache



Bei einem Plattenspeicherzugriff werden meistens 4096 Bytes große Blöcke von Daten transportiert (identisch mit der Rahmen Größe).

Ist kein Cache in der Coupling Facility vorhanden, erfolgt der Zugriff auf die folgende Weise:

1. **Lock auf die gewünschte Seite setzen.**
2. **Seite lesen, falls im Buffer Pool vorhanden**
3. **Falls nicht vorhanden, Seite vom Plattenspeicher einlesen.**
4. **Update des Cache Directories in der CF.**

Wird ein Coupling Facility Cache genutzt, wird die Seite aus dem Cache gelesen, falls vorhanden. Wenn nicht, wird die Seite vom Plattenspeicher eingelesen.

In diesem Fall schreibt DB2 zusätzlich die Seite in den CF Cache. Dies ist ein Store-in-Cache; nicht bei jedem Update der Seite wird diese auf dem Plattenspeicher geschrieben. Somit kann die CF Cache Version des Blockes jüngeren Datums sein als die Version auf dem Plattenspeicher.

Adabas

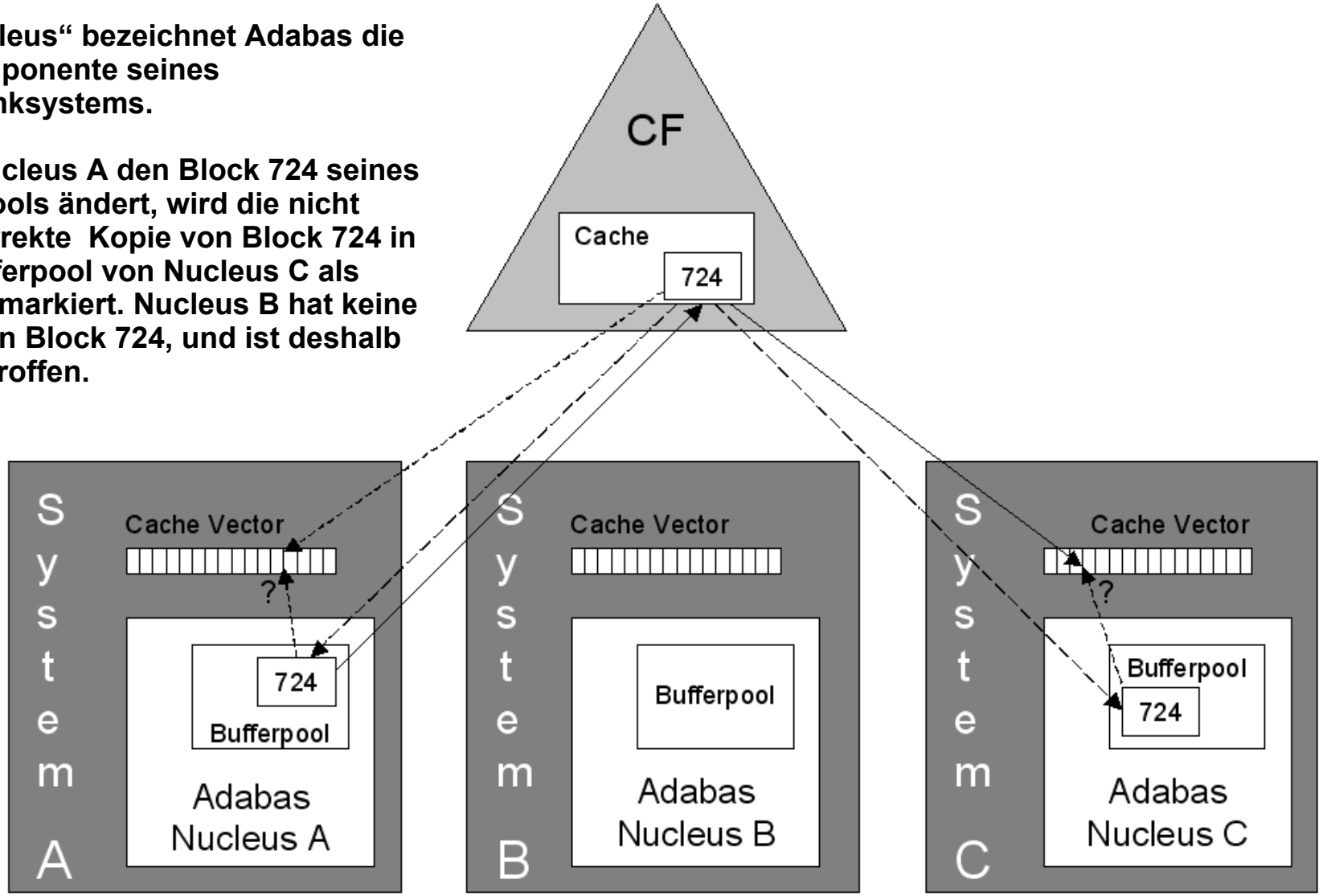
Adabas (Adaptable Database System) wurde von der Software AG, Darmstadt, in 1971 eingeführt. Ursprünglich war ADABAS nur auf Mainframes verfügbar. Heute läuft ADABAS unter den z/OS, VSE, VM/CMS, Fujitsu's Mainframe Betriebssystem Facom, den Fujitsu/Siemens BS2000 Betriebssystemen sowie den Windows, Solaris, AIX, HP-UX, SUSE-Linux und Red Hat Linux Betriebssystemen.

Weltweit wird Adabas von etwa 3000 Installationen benutzt, darunter etwa 100 – 200 z/OS Installationen. Adabas ist ein Geheimtipp für neue Datenbankbenutzer; in Bezug auf Performance nimmt es auch heute noch eine Spitzenposition ein.

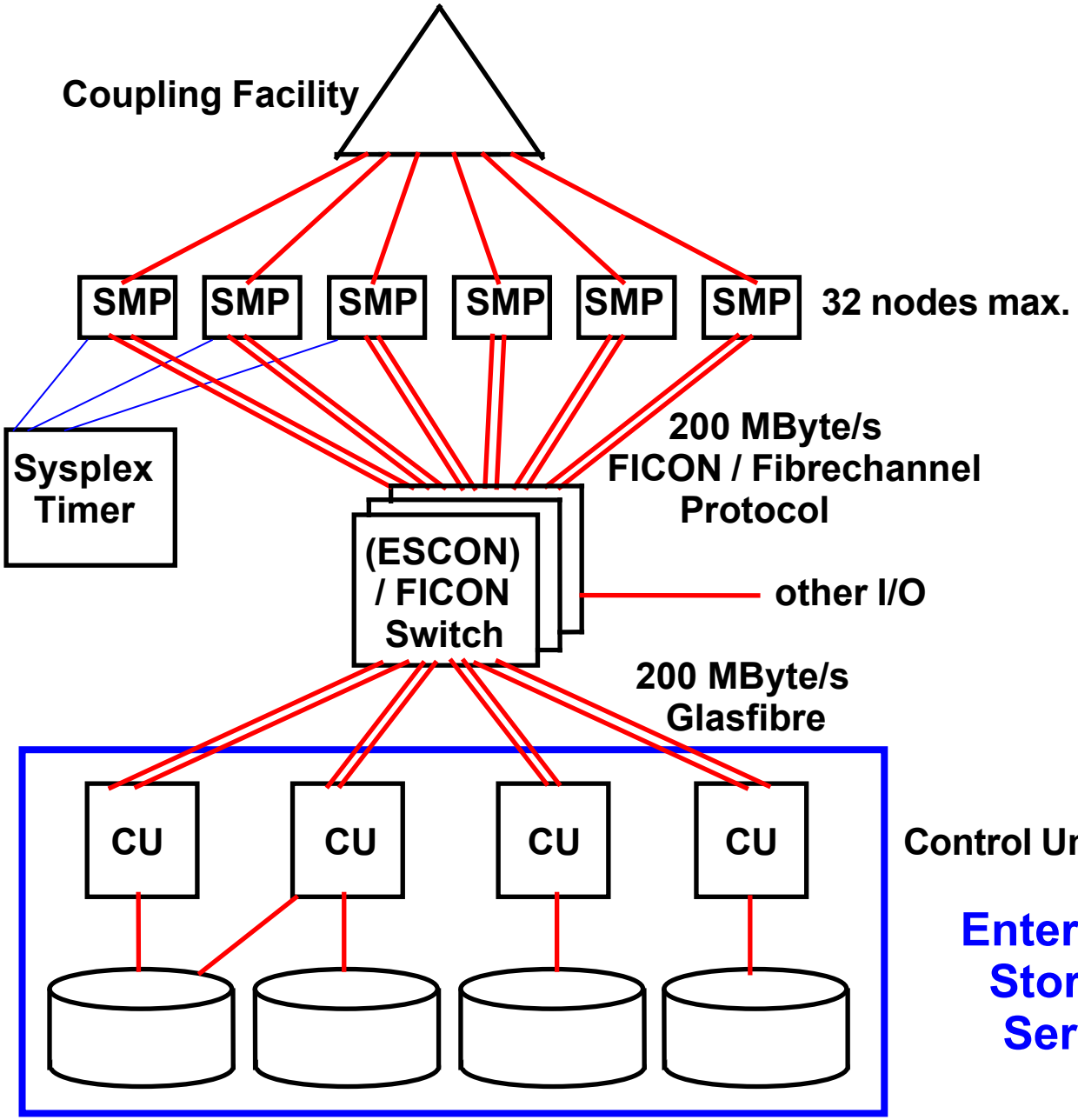
Adabas ist kein relationales Datenbanksystem (auch wenn die Firma Adabas behauptet, Adabas sei fast relational). Daten in der ADABAS-Datenbank sind hierarchisch in Files strukturiert; eine File entspricht in etwa einer SQL-Tabelle. Die Files bestehen aus Records; die Felder der Records entsprechen etwa den Spalten einer SQL-Tabelle. Es bestehen einige Ähnlichkeiten mit der IMS-Datenbank.

Als „Nucleus“ bezeichnet Adabas die Kernkomponente seines Datenbanksystems.

Wenn Nucleus A den Block 724 seines Buffer Pools ändert, wird die nicht mehr korrekte Kopie von Block 724 in dem Bufferpool von Nucleus C als ungültig markiert. Nucleus B hat keine Kopie von Block 724, und ist deshalb nicht betroffen.



Die Adabas Database der Software AG in Darmstadt nutzt ebenfalls die Coupling Facility als Plattenspeicher-Cache.



Problem: Die Coupling Facility ist nur mit den Knoten (Systemen) des Sysplex verbunden; sie hat keine direkte Verbindung mit den Plattenspeichern.

Frage: Wenn der Plattenspeicher-Cache in der Coupling Facility zu voll wird, wie werden Teile auf einen Plattenspeicher ausgelagert, um Platz zu schaffen?

Enterprise Storage Server

Cast Out

Wird ein neuer Buffer in die CF Cache geschrieben, muss dafür Platz geschaffen werden und ein anderer CF Puffer auf den Plattenspeicher ausgelagert werden. Die Coupling Facility ist aber nur mit den Systemen (Knoten) verbunden. Sie hat keinen direkten Zugriff auf die Plattenspeicher.

DB2-Instanzen in den einzelnen Systemen unterhalten jeweils einen „Cast-Out“ Thread, der einen Buffer aus dem Cache lesen und auf den Plattenspeicher schreiben kann.

Die Cast-Out-Verantwortung wird nach dem Round-Robin-Algorithmus (oder einem anderen Algorithmus) den einzelnen Threads zugeordnet. Ein Cast-Out erfolgt jeweils für eine Gruppe von Seiten.

CF Cache Recovery

Frage: Ist es nicht bedenklich, Daten nur in den Coupling Facility Cache zu schreiben, und nicht sofort auf den Plattenspeicher?

Ein Duplikat aller Daten befinden sich in den Buffer Pools der einzelnen Systeme. Im Fehlerfall kann hiermit ein Rebuild des CF Cache-Inhaltes stattfinden. Weiterhin verfügt eine Installation praktisch immer über 2 CFs, wobei die zweite CF alle Daten der ersten CF dupliziert und im Fehlerfall automatisch die Funktionen der ersten CF übernehmen kann.

CF Listen- / 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:

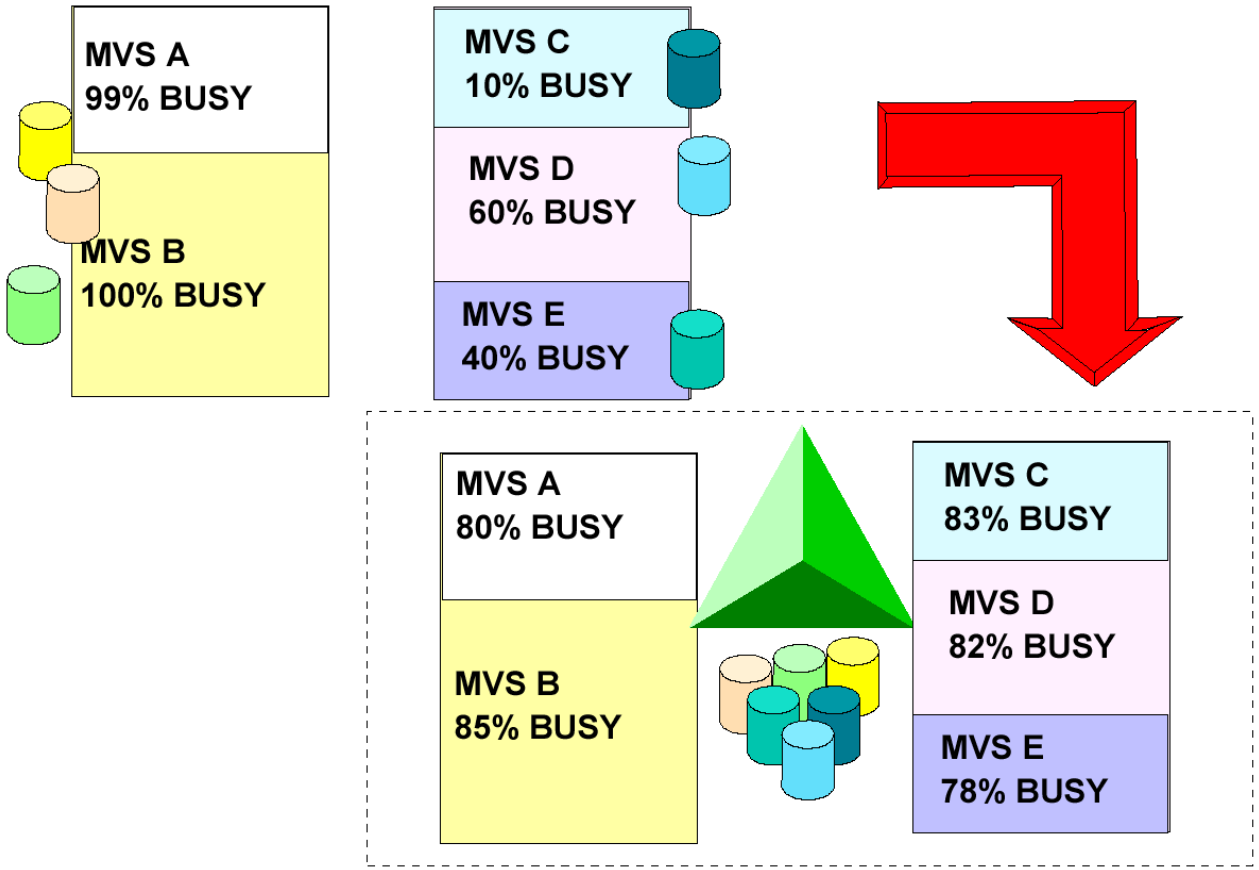
- Clusterweite RACF-Steuerung. Ein Sysplex-Cluster besteht aus mehreren z/OS-Instanzen. Im einfachsten Fall müsste sich ein Benutzer mit getrennten Passworten in jedes System einzeln einloggen. „Single Sign On“ ist eine Einrichtung, mit der der Benutzer mit einem einzigen Sign On Zugriffsrechte auf alle Ressourcen 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 Statusinformationen aus, um Transaktionen dynamisch an unterbelastete Systeme weiter zu reichen.
- MQSeries: Queue-Sharing Group eines WebSphere MQ-Clusters

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

- LIFO Queue
- FIFO Queue
- Key Sequenced

Key Sequenced bedeutet, dass auf ein bestimmtes Item in einer Queue/List-Struktur mittels eines Schlüssels zugegriffen werden kann.

Work Load Balancing

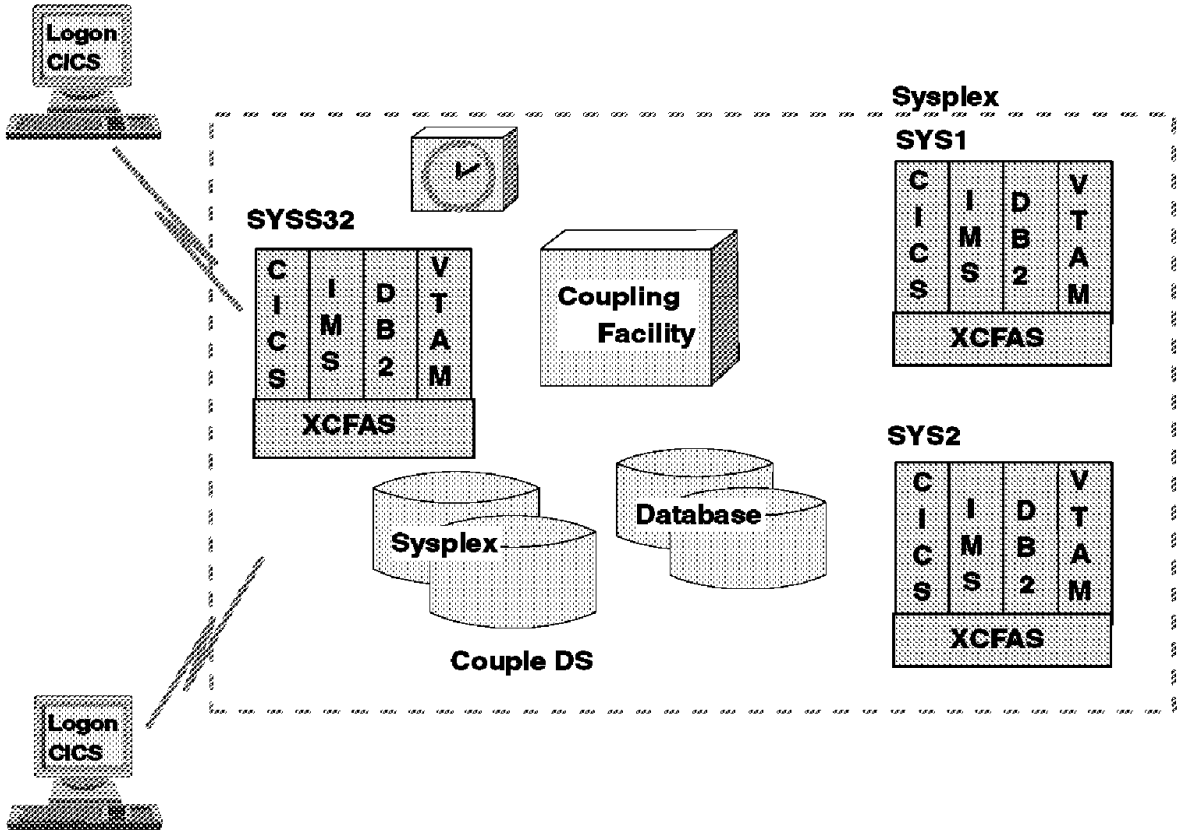


Gezeigt sind 5 Systeme: MVS A, MVS B, MVS C, MVS D und MVS E.

Es ist fast unausbleiblich, dass diese Systeme unterschiedlich ausgelastet sind.

Indem man die Rechner zu einem Sysplex zusammenfasst, kann eine Systemkomponente, der „Work Load Manager“ (WLM) für eine gleichmäßige Auslastung aller 5 Systeme sorgen. Die entsprechenden Daten werden in der CF gespeichert.

Cross Coupling Facility Address Space XCFAS



Angenommen seien mehrere Instanzen einer Anwendung oder eines Subsystems auf unterschiedlichen Knoten eines Sysplex, z.B. CICS oder WebSphere. Mit Hilfe von XCFAS können die Instanzen Statusinformationen 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-Performance

... und wie sieht es mit dem Leistungsverhalten und der Skalierbarkeit eines Sysplex aus?

In den meisten Fällen ist es sehr schwierig, eine existierende Anwendung, die auf einer einzelnen CPU läuft auf einen Mehrfachrechner zu portieren. In vielen Fällen bringt bei 10 oder 30 CPUs jede weitere CPU keinen nennenswerten Leistungsgewinn mehr.

Der Sysplex und besonders die Coupling Facility bilden hier eine Ausnahme. Sie weisen hervorragende Skalierungseigenschaften auf. Dies wird in den beiden folgenden Abbildungen dargestellt.

Heute (2013) ist die Coupling Facility immer noch ein Mainframe-Alleinstellungsmerkmal. Wir erwarten, dass in Zukunft die Geschwindigkeit einer einzelnen CPU nur noch langsam steigen wird. Leistungssteigerungen unser Rechner sind daher in Zukunft in erster Linie durch den Einsatz von Mehrfachrechnern zu erwarten.

Es ist daher vorstellbar, dass eine Coupling Facility in Zukunft auch für die x86 Plattform verfügbar sein wird – warten wir es ab.

Sysplex Overhead

Installation	Anzahl Systeme	% Sysplex Overhead
A	4	11 %
B	3	10
C	8	9
D	2	7
E	11	10
DB2 Warehouse Workload	2	13

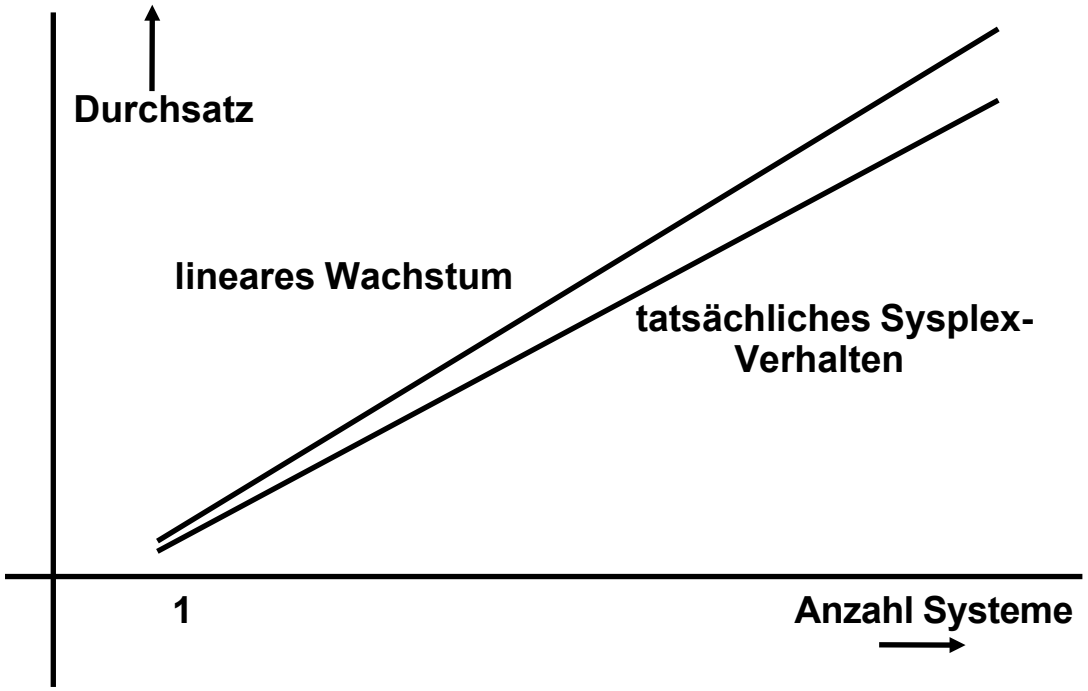
Gezeigt ist der durch die Sysplex Support Software verursachte Overhead (zusätzlich benötigte CPU Zyklen). Die Messungen erfolgten auf den Sysplex Installationen von 5 großen Unternehmen, hier als A, B, C, D und E bezeichnet. Weiterhin erfolgten Messungen auf einer IBM internen DB2 Warehouse Installation.

Der Overhead bewegt sich zwischen 7 und 13 %.

Die Sysplex Support 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.

Die Sysplex Support Software wird nur dann installiert, wenn der Rechner als Bestandteil eines Sysplex genutzt wird.

Sysplex Overhead

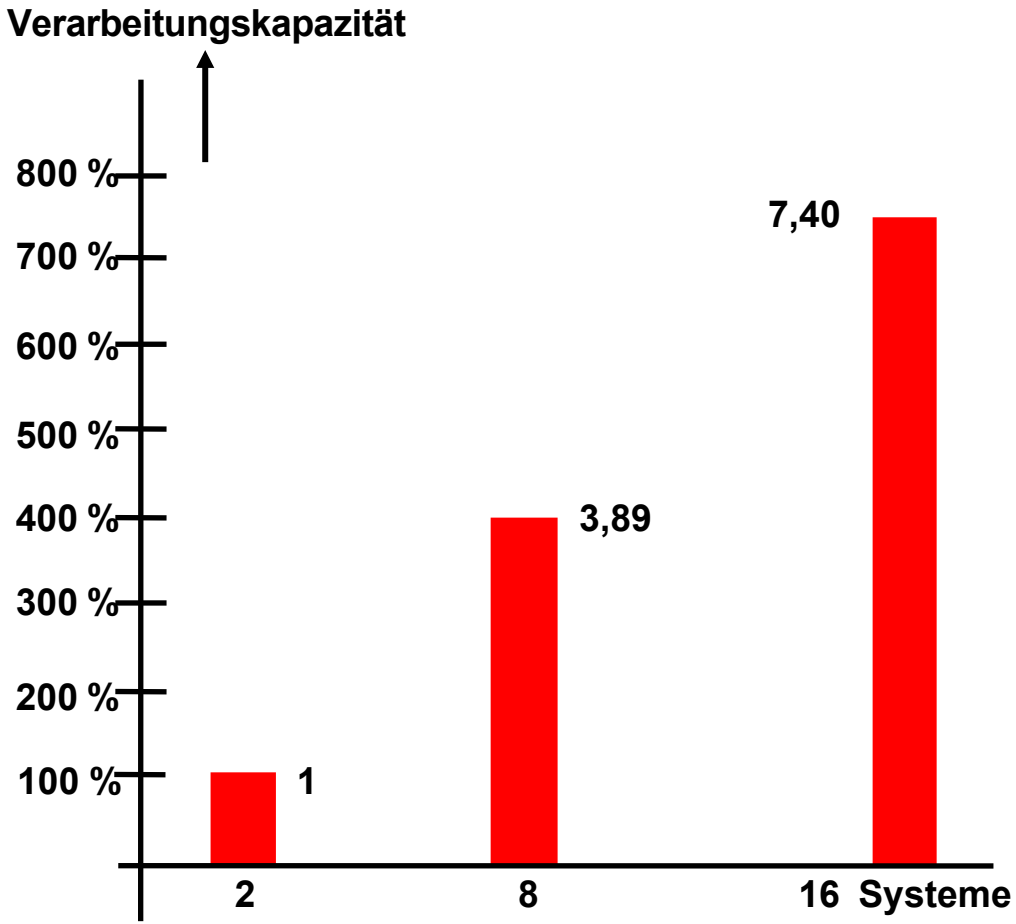


Mit nur einem einzigen System entsteht der erwähnte Sysplex Support Software Overhead. Mit einer wachsenden Anzahl von Systemen beobachten wir eine lineare Skalierung.

“The Parallel Sysplex environment can scale nearly linearly from 2 to 32 systems. The aggregate capacity of this configuration meets every processing requirement known today.”

http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp?topic=/com.ibm.zos.zconcepts/zconc_pllsysscale.htm

Parallel Sysplex Leistungsverhalten



Mit 2 Systemen, die mit einer CF verbunden sind, erreicht man eine Leistung von 100 %.

Mit 16 Systemen sollte man theoretisch eine Leistung von 800 % erreichen. Tatsächlich erreicht werden 740 %.

Jedes System kann z. B. 8 oder 16 CPUs enthalten.

Dies ist ein sehr guter Skalierungswert.

Testergebnisse einer Installation bestehend aus CICS-Transaktionsmanager, CICSplex System Manager und einer IMS Datenbank, mit einer Mischung von OLTP, Reservierung, Data Warehouse und Bankanwendungen.

Literatur: Coupling Facility Performance: A Real World Perspective, IBM Redbook, March 2006, http://www.redbooks.ibm.com/redpapers/pdfs/redp4014.pdf

Sysplex Teil 5

Weiterführende Informationen

Literatur

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

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

Sonderheft des IBM Journal of Research and Development, Vol. 36, No.4, July 1992 zum Thema Sysplex Hardware:

Sonderheft des IBM System Journal, Vol. 36, No.2, April 1997 mit folgenden Aufsätzen:

J. M. Nick, B. B. Moore, J.-Y. Chung, and N. S. Bowen: S/390 cluster technology: Parallel Sysplex, p. 172

N. S. Bowen, D. A. Elko, J. F. Isenberg, and G. W. Wang: A locking facility for parallel systems, p. 202

G. M. King, D. M. Dias, and P. S. Yu: Cluster architectures and S/390 Parallel Sysplex scalability, p. 221

J. W. Josten, C. Mohan, I. Narang, and J. Z. Teng: DB2's use of the coupling facility for data sharing , p. 327

T. Banks, K. E. Davies, and C. Moxey: The evolution of CICS/ESA in the sysplex environment, p. 352

J. P. Strickland: VSAM record-level data sharing, p. 361

Eine gut gemachte Video Präsentation zum Thema Sysplex ist zu finden unter:
http://www.yourepeat.com/watch/?v=OGJBq_s-a9A&feature=youtube_gdata

Ein weiteres Video erläutert, wie DB2 die Coupling Facility nutzt um Data Sharing zu implementieren:
http://www.yourepeat.com/watch/?v=OGJBq_s-a9A&feature=youtube_gdata

Information über das Sysplex Leistungsverhalten

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.92.5684
oder
http://www-sr.informatik.uni-tuebingen.de/~bloching/papers/pdpta00.pdf

Eine schöne Zusammenfassung über Mainframe Kurzinformationen ist zu finden unter
http://idcp.marist.edu/enterprisesystemseducation/ztidbitz.html

Das Marist College im Staate New York (http://www.marist.edu/) ist eine der weltweit führenden Hochschulen auf dem Gebiet der akademischen Mainframe Ausbildung.

Die neue (2013) Sun-Oracle Produktlinie ist beschrieben in

http://www.theregister.co.uk/2013/03/26/oracle_sparc_t5_m5_server/
und
http://www.theregister.co.uk/2013/03/26/oracle_sparc_t5_m5_server/page3.html