IMS in a Parallel Sysplex

Enhancing Availability, Capacity, and Workload Balancing

The world depends on it

Rich Lewis and Bill Stillwell
IBM Dallas Systems Center

Trademarks

- **The following are registered trademarks of International Business Machines Corporation**
  - ACF/VTAM
  - AIX
  - APPN
  - DB2
  - IBM
  - MVS
  - OS/390
  - Parallel Sysplex
  - S/390
  - VTAM
  - WebSphere
- **The following are trademarks of International Business Machines Corporation**
  - CICS
  - IMS
  - MVS/ESA
- LINUX is a registered trademark of Linus Torvalds and others.
- Windows is a trademark of Microsoft Corporation.
- Solaris is a trademark of Sun Microsystems, Inc..

Abstract and Agenda

Abstract

- IMS uses Parallel Sysplex to deliver improved availability, increased capacity, and workload balancing. This presentation shows how IMS data sharing, VTAM generic resources, shared queues, automatic restart management (ARM), and other facilities deliver the promise of Parallel Sysplex

Agenda

- Availability, Capacity, and Workload Balancing
- IMS Database Manager use of Parallel Sysplex
- IMS Transaction Manager use of Parallel Sysplex
- Parallel Sysplex Failure Recovery

This presentation explains how IMS exploits the Parallel Sysplex to improve availability, capacity, and workload balancing. At the end there is a list of Redbooks and articles available on the Web which provide more detailed information. There is also a list of acronyms and terms used in this presentation.

Goals - Improved ...

- **Availability**
  - Access, by the end user, to the data and facilities of IMS necessary to perform that end user's business function

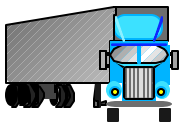
  - Parallel Sysplex allows us to have multiple systems (clones) to help us survive outages
    - The clones must have access to the same data and the capability to do the same work
  - High availability requires an ability to survive both planned and unplanned outages
  - High availability requires an ability to quickly restore failed systems

High availability implies that the end user has access to the facilities and resources needed to perform some business function when it is needed. Parallel Sysplex allows us to have multiple systems which can do the same work. These systems are called clones. The clones must have access to the same data and the capability to process the same transactions or work requests. High availability requires that we be able to survive both planned outages, such as upgrades, and unplanned outages, such as subsystem abends. Even with clones, we need the capability to quickly restore any failed system.

Goals - Improved ...

Capacity

- The ability of the IMS subsystems to process the total workload within the required time



- Parallel Sysplex gives us the capability to add (or subtract) capacity more easily
  - We can add (or delete) clones
- Using the capacity requires an ability to route the work to the system(s) with available capacity

Adequate capacity implies that the servers have the resources to satisfy all of the work requests from its end users within the needed time frame. Parallel Sysplex allows us to meet capacity needs by easily adding and deleting clones. We when change our configuration in this way, we need to be able to route the work to the systems which can handle it.

Goals - Improved ...

- **Workload Balancing**
  - The spreading of work across the systems which can do it

  - Parallel Sysplex allows us to automatically balance workloads
  - Balancing allows us to more easily handle spikes in demand
  - We need to adjust to workloads and system configurations

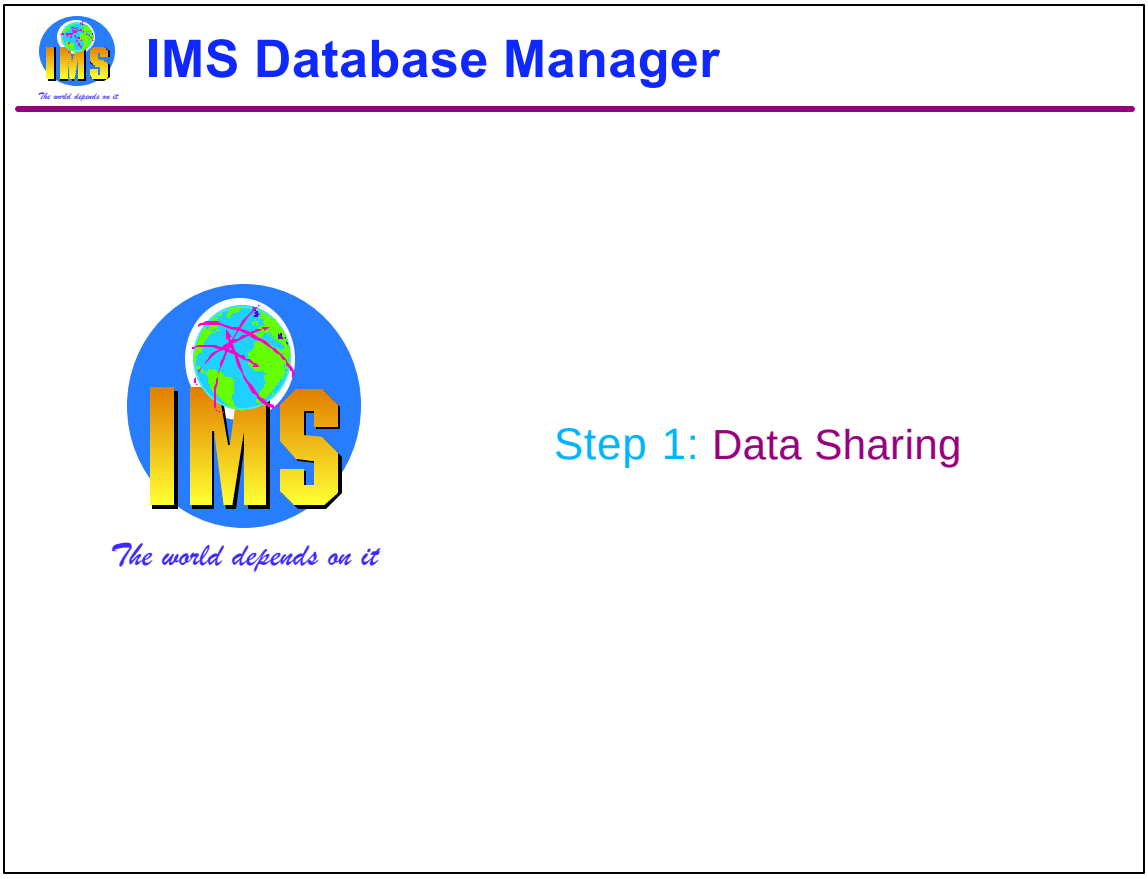
Workload balancing is the spreading of work across the systems which can do it. It also requires the work to be spread appropriately. Parallel Sysplex provides capabilities to automatically balance workloads. Proper balancing allows us to more easily handle unexpected peaks in workloads. For balancing to work best, it must be able to dynamically adjust to changes in the workload and changes in our hardware and software configurations.

Parallel Sysplex

Parallel Sysplex provides facilities for

- Increased availability
- Enhanced capacity
- Improved workload balancing

We will see how IMS and other products, when used in a Parallel Sysplex, provide facilities to increase our availability, enhance our capacity, and more easily balance our workloads.

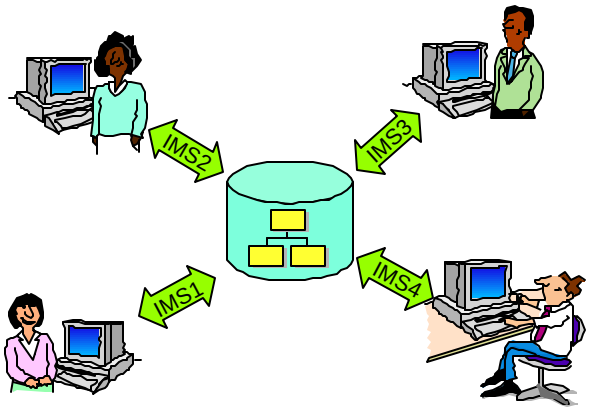


The first step in using Parallel Sysplex with IMS is the implementation of data sharing by the database manager. Data sharing allows multiple IMS subsystems to access and update the same IMS databases.

Block Level Data Sharing

Block level data sharing (BLDS)

- N-way data sharing for databases
  - Up to 255 IMS subsystems on 32 MVSs
- Full capabilities
  - Multiple updaters
  - Data integrity

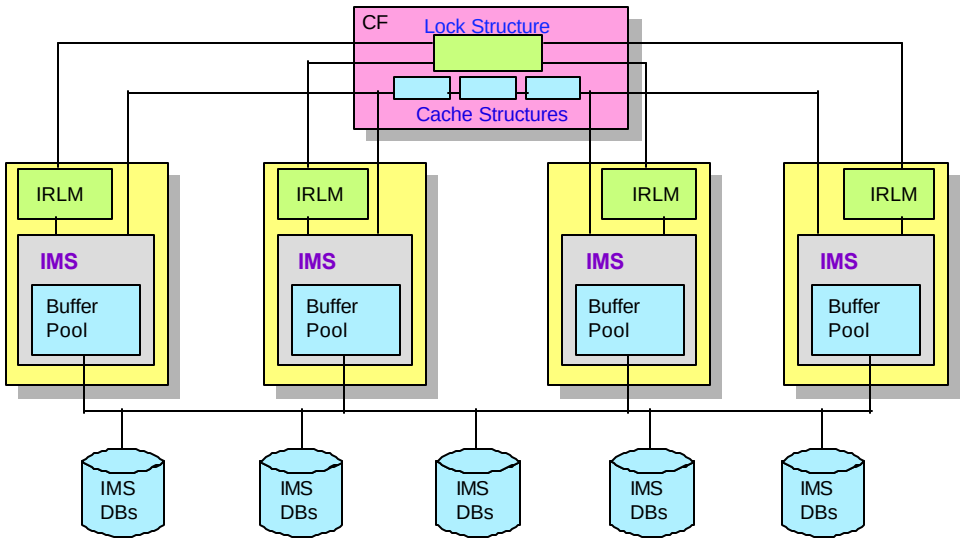


Block level data sharing means that each IMS server has the authority to access and update all shared data. As many as 255 IMS subsystems on up to 32 MVS (OS/390 or z/OS) LPARs or processors can share IMS databases. An IMS subsystem is either an online system or a batch (DLI or DBB) job.

Each IMS has full capability for access and update with integrity.

Parallel Sysplex data sharing was introduced in IMS Version 5.

BLDS Configuration



IMS systems include TM/DB, DBCTL, and IMS batch jobs.

This chart shows a data sharing configuration with 4 IMSs running on 4 MVS images sharing the same set of IMS databases. Each IMS system has its own database buffer pools. Each IMS reads and updates the databases. To support the integrity requirements of this configuration, IMS utilizes structures in Coupling Facilities. A lock structure is used to hold locking information which is shared by the lock managers (IRLMs) used by the IMS systems. Information about database blocks and their locations in the buffer pools is stored in cache structures. These are used to maintain the integrity of the buffer pools when an IMS updates a block.

IMS Data Sharing

- **No IMS DB restrictions on applications which may use data sharing**
  - All full function databases may be shared
  - All Fast Path DEDBs may be shared
  - MSDBs may be replaced with DEDB VSO
- No IMS data affinities
  - All IMS data may be available to all IMSs

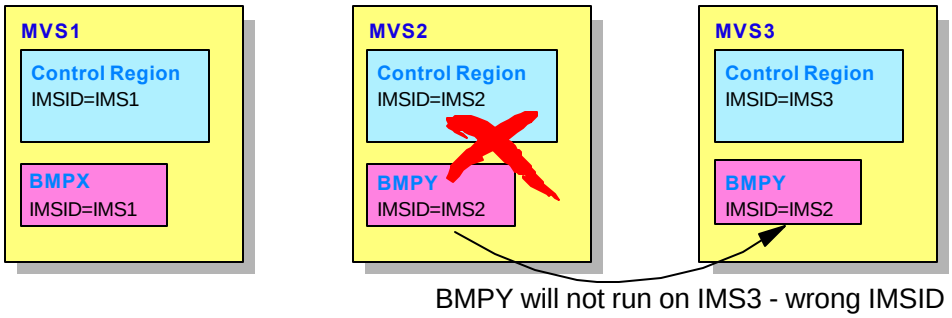
As of IMS V6, there are no restrictions on what IMS data can be shared. If you are using Fast Path MSDBs, they must be converted to DEDBs. The VSO option for DEDBs can provide performance similar to MSDBs.

Since all IMS data may be shared, IMS does not force data affinities on an installation. A data affinity occurs when some data, such as a database, is not shared. Its access can only occur from one system. Without data affinities, a transaction or batch process is capable of running on any system. It does not have to be routed to a particular IMS because that's the only one with access to the data.

Dependent Region Movement

Movement of a dependent region across IMSs

- Dependent regions specify the IMS to which they will connect
  - IMSID parameter
- Each IMS has a unique IMSID
- Could make movement of BMPs to different IMS difficult
  - Movement may be needed if an IMS fails

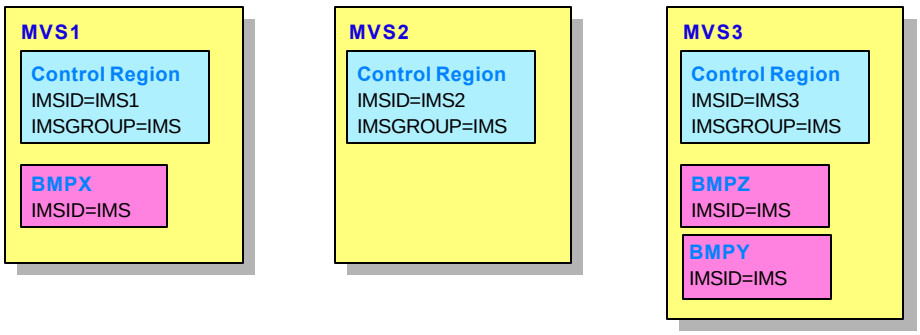


Without Parallel Sysplex we are used to running a dependent region with only one control region. We do not move a dependent region from one IMS to another. With Parallel Sysplex we may want this ability, especially for BMPs. We use the IMSID parameter in the dependent region to specify to which control region the dependent region will connect. Each IMS has a unique IMSID. This makes the movement of a dependent region, such as a BMP, difficult. It seems we have to change the JCL to specify a different IMSID before we can move the dependent region. In this example, MVS2 fails. We want to execute BMPY with IMSID=IMS2 specified on MVS3 where IMS3 is running. This will not work. But, there is a solution.

IMSGROUP for Dependent Regions

IMSGROUP for BMPs, MPPs, and IFPs

- Generic name for IMS Control Regions (IMSGROUP)
  - Dependent region may connect to any control region with this generic name (IMSID=generic name)
  - BMP availability, capacity, and workload balancing are made easier
- If IMS2 not available, BMPY can run with IMS1 or IMS3



The IMSGROUP parameter allows IMS to be known by both its IMSID and its IMSGROUP name. All the IMSs must have unique IMSIDs, but they may have the same IMSGROUP name. They register this group name using MVS token services and their own unique IMSID. The BMP has an IMSID equal to the IMSGROUP name. The BMP uses token services to see if there is an IMSID using the "IMS" as an IMSGROUP parameter. In the example, BMPY finds that it is running on the same MVS with IMS3. It would connect to IMS3. In this way, if IMS2 is down, or that system is very busy, the BMP could be routed to any MVS which has any IMS in the group.

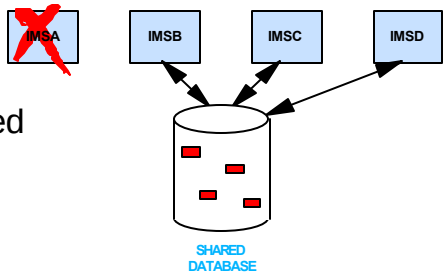
The Program Restart Facility (5655-E14) may be used to easily restart a failed BMP on another IMS system. This further extends the capabilities to use BMPs with Parallel Sysplex.

Fast Database Recovery (FDBR)

FDBR enhances availability

The problem:

- Active IMS systems acquire database locks
  - Update locks held in IRLM lock structure in CF
- If IMS system fails
  - Locks for in-flight updates are *retained* until emergency restart
- Other IMS systems cannot access locked data
- Requests to access data locked by failed system result in *application ABENDs*



Fast Database Recovery (FDBR) was introduced in IMS Version 6. It may be used to greatly reduce the effect of the failure of an IMS system on data availability to other IMS systems.

Applications which run in IMS systems acquire locks. These locks provide data integrity. Locks which potentially protect updates are kept in a coupling facility lock structure. If an IMS system fails, these update locks are retained. They must be kept until the inflight work of the failed system is backed out. Without FDBR this is done during emergency restart. Locked records cannot be accessed by other IMS systems. These systems do not wait for the release of the locks. Instead, their applications get a lock reject condition when they ask for a lock which is retained for the failed system. This lock reject condition typically causes an application abend. So, the failure of one IMS system affects the other IMS systems. They do not have access to some data until the inflight transactions are cleaned up.

Fast Database Recovery (FDBR) ...

The solution:

- FDBR is an independent region which tracks an active IMS
  - FDBR region may be on same or another MVS
  - FDBR and IMS join same XCF Group for monitoring
  - FDBR reads IMS's logs
- When IMS fails, FDBR is invoked
  - Failure detected by XCF notification or reading IMS failure log record (x'06')
- FDBR restores databases to last point of consistency
  - DL/I dynamic backout
  - DEDB redo processing
- FDBR purges retained locks from IRLM
- FDBR is releases locks much faster than emergency restart

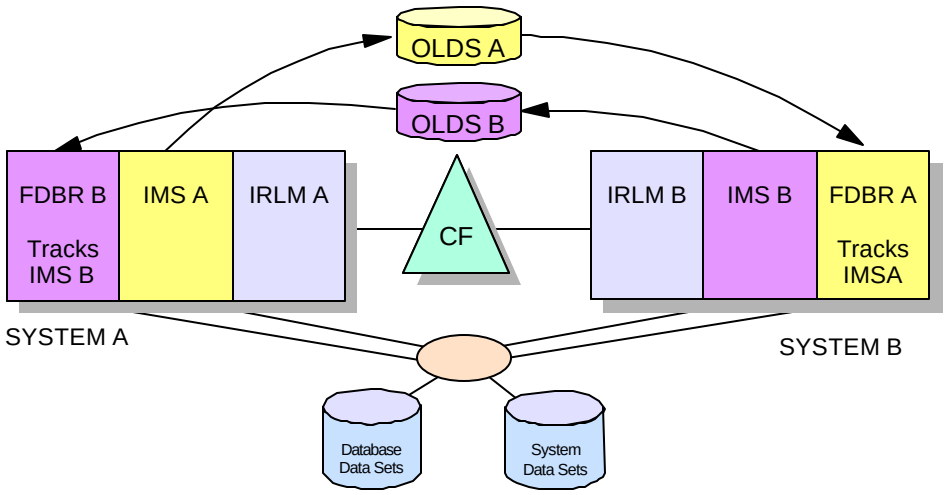
FDBR is the solution for the problem mentioned on the previous page. FDBR is an independent region. It runs in its own address space. An FDBR region tracks one IMS control region. For maximum effectiveness, an FDBR should execute on a different system from the one where the IMS which it is tracking runs.

Tracking is done in two ways. First, an FDBR and its IMS system join the same XCF group. This allows FDBR to be immediately aware when the tracked IMS's address space or MVS terminates. Second, FDBR continually reads the tracked IMS's log (OLDS). If IMS abends, its ESTAE routine writes a failure log record (type x'06'). FDBR may read this log record before IMS's address space terminates.

When either of these tracking methods makes FDBR aware of the IMS failure, FDBR restores the databases to the last point of consistency. For full function databases, this means it backs out inflight units of work. For DEDBs, this means that it invokes redo processing for incomplete output threads. These are the same actions that emergency restart would have done. When these actions are complete, FDBR releases the locks held by the failed IMS. Once again, this is what emergency restart would have done.

FDBR is much quicker than emergency restart. FDBR does not have to wait for the restart of IMS. It does not have to wait for the loading of the IMS modules. It does not have to wait for the reading of the log except the last few records. Since FDBR is much quicker, it eliminates many of the potential lock rejects and application abends on the surviving IMSs.

Fast Database Recovery (FDBR) ...



If SYSTEM A or IMS A fails,
FDBR A backs out all in-flight work from IMS A
IMS B has access to all IMS data

For maximum effectiveness, FDBR should not run on the same MVS where its IMS is running, otherwise an MVS failure would cause both to fail. This illustrates a potential configuration. IMS A and its FDBR run on different systems. IMS B and its FDBR run on different systems. If SYSTEM A or IMS A fails, FDBR A backs out all of the in-flight work from IMS A. It also releases the retained locks held for the in-flight work of IMS A. This allows IMS B to access all of the IMS data.

IMS Database Summary

IMS data sharing in a Parallel Sysplex

- Higher availability
- Increased capacity
- Usable by IMS TM, CICS, and IMS batch

Provides access to data from multiple IMSs

- All IMS data may be shared
- Enables IMS cloning

Higher availability is provided by the data sharing configuration. It allows work which might otherwise have to wait for the restart of a failed IMS to run on a surviving IMS. FDBR reduces the impact of the failure by monitoring an active IMS and performing dynamic backout and/or DEDB redo processing sooner than an IMS emergency restart would do.

Because data sharing enables multiple IMSs to access the same data, work can execute on any and all IMSs in the sysplex data sharing group. As many as 32 processors can be used to provide maximum capacity.

If every IMS can access all the data, then every IMS can process any of the work. This allows an installation to create IMS clones. That is, each IMS system definitions can be the same. In fact, a single system definition can be used for all the IMSs. Cloning allows us to distribute the work to the systems which have the capacity to handle it. We will see how this works in the next parts of this presentation.

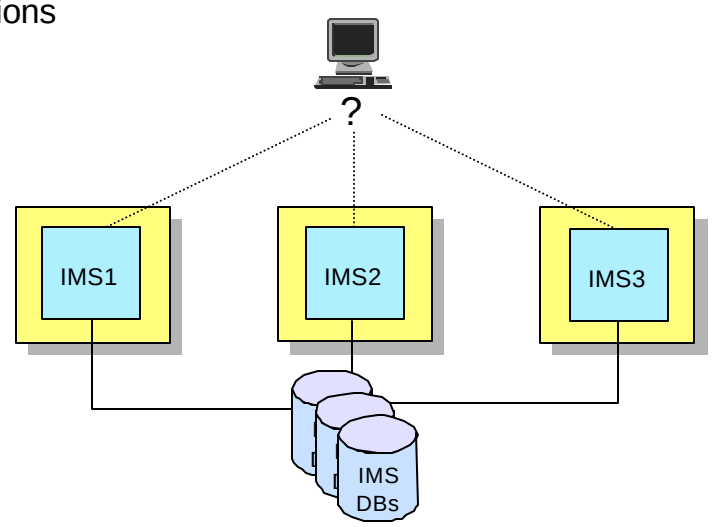
Since we are using data sharing, all IMSs have access to the data. This allows us to do our work on any IMS.

The second step in using Parallel Sysplex with IMS Transaction Manager is the distribution of connections. For VTAM, connections are sessions. For TCP/IP, they are socket connections. Distributing connections is one of the methods for distributing our workload across multiple IMSs and multiple processors. This is static distribution. That is, once a user is connected to an IMS, the user remains connected until the connection is broken. Another connection is required to use this method for distributing the workload to another IMS.

Distributing Transaction Workload

How do we distribute the workload?

- Distribute logons (connections)
- Distribute transactions
- Combination



There are several alternatives to distributing the transaction workload. Two basic techniques are:

1) Distribute the logons so that not all users are logged on to the same IMS. Whichever IMS they are logged on to is the one that processes the transaction.
2) Distribute the transactions between IMSs once it has been received from the network.

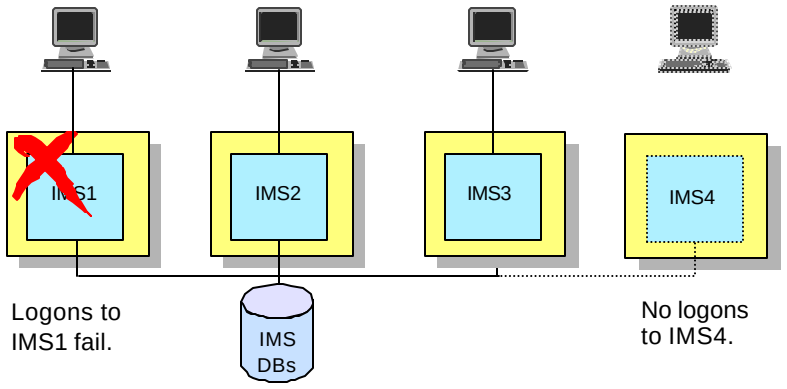
And there is, of course, a combination whereby users logons are distributed and the transaction submitted by these users are also distributed after they are entered.

Because we are data sharing, it doesn't matter where the transaction ends up being processed.

Distributing Logons

Distributing logons manually

- Tell each user which IMS to logon to
- Not dynamic
  - Balancing logons responsibility of user admin
  - Problems when 1 IMS down, new IMS added, new users, ...



The earliest approach to distributing logons was to just tell the end user which IMS to log on to. There are several problems with this technique, however.

1) Balancing the logons becomes an administrative responsibility which must be monitored continuously as users come and go.
2) As new IMSs join the group, either no users log on to that IMS, or the administrator must reassign users to IMSs.
3) If an IMS fails, users have to be instructed either to wait for it to restart or to log on to another IMS. Once a user knows of another IMS, he may decide arbitrarily to log on to it instead of his primary IMS, defeating the balancing goal.

Distributing Logons Automatically

Distributing logons (connections)

- SNA
  - VTAM Generic Resources
  - USERVAR exit
- TCP/IP
  - DNS/WLM (Domain Name Server/Workload Manager)
  - IND (Interactive Network Dispatcher)
  - Sysplex Distributor

Distributing logons can be done automatically using several techniques.

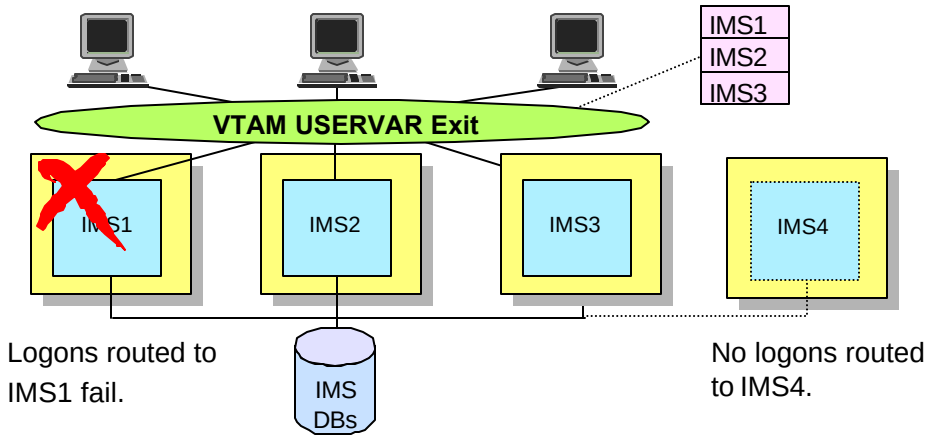
For SNA networks, VTAM Generic Resources can be used to dynamically route a logon request to an active IMS. The IMS is chosen based on Workload Manager (WLM) information or the number of users currently logged on. This capability is available with IMS Version 6 and later releases. Users of IMS Version 5 may use a VTAM USERVAR exit. This exit can be used to direct the logon request to one of several IMSs in the group.

For TCP/IP, connection distribution can be accomplished using such tools as DNS/WLM, IND, or the Sysplex Distributor.

VTAM USERVAR Exit

VTAM USERVAR exit

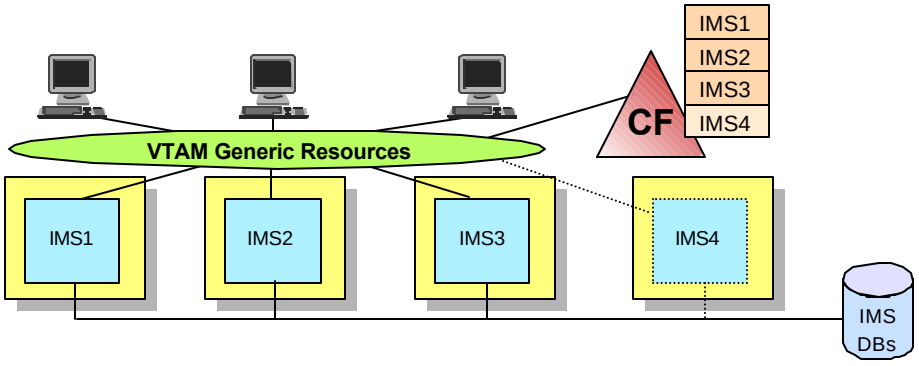
- Exit routine used to route logon to any IMS
- Not dynamic
  - Exit routine is not aware of changes in configuration or availability



USERVAR is a VTAM capability that can change the value specified for the VTAM application name in a logon request. USERVAR support includes an optional exit routine. The exit routine can choose from multiple application names. So, a USERVAR exit routine can be used to route a logon to any IMS that it knows about. But, it may not know of configuration changes or of the availability of any particular IMS in the group. For example, if IMS1 fails, the exit routine might continue to route logons to IMS1. Similarly, if IMS4 is added to the configuration, the exit routine might not route any logons to it. A sophisticated routine might be able to modify its decisions, but there is no automatic notification to the routine of changes in the configuration.

VTAM Generic Resource

- All IMSs join a Generic Resource Group
- User logs on to generic resource name
- VTAM selects available IMS from members of the Group
  - Failed IMS systems removed from group dynamically
  - New IMS systems added to group dynamically



VTAM Generic Resources (VGR) is a service provided by VTAM in a Parallel Sysplex. It minimizes the knowledge that an end user needs to log on to one of several like instances of an application, such as IMS. Each instance of an application, joins a Generic Resource Group by specifying both the group name and its specific VTAM application name. End users specify the group name when logging on. VTAM selects a member of the group for the session.

Generic Resource Groups are dynamic. When a new IMS opens its VTAM ACB, it joins the group and is a candidate for subsequent logons. When an IMS terminates, it is removed from the group. It is then no longer a candidate for logons.

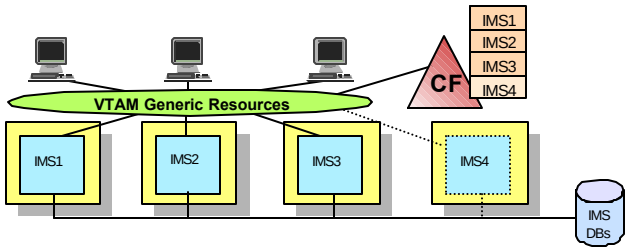
Information about the members of a group is kept in a Coupling Facility structure.

APPC/IMS may use VGR, but this does not require direct IMS support. Instead, this support is provided by APPC/MVS.

VTAM Generic Resources ...

VTAM Generic Resources Benefits

- Availability
  - User logon request routed to any available IMS
  - User does not need to know what's available
- Capacity
  - New IMS systems may be added without changes to user procedures
- Workload (logon) Balancing
  - Spreads users across multiple IMS systems



There are many benefits of VGR over other techniques for distributing logon requests.

1) Availability. VTAM knows by looking in the CF Structure which IMSs are active. It routes requests only to active IMSs. If an IMS fails, its users can immediately log on again using the same generic name. They will be connected to one of the surviving IMSs.

2) Capacity. If another IMS is needed to handle the workload, it immediately becomes eligible for user logons. User procedures do not have to be modified.

3) Workload Balancing. VTAM attempts to balance logons across the available IMSs. It has two ways of doing this. First, if Workload Manager (WLM) goal mode is used, VGR routes a logon to the system with the most available capacity. Second, if WLM goal mode is not used, VGR attempts to balance the number of logons per IMS system. Users may implement a VGR user exit routine to override the VGR decision.

Web and TCP/IP Connections to IMS

Most typical Web Server and TCP/IP connections to IMS

- APPC
  - May use VTAM Generic Resources for APPC/MVS
- TCP/IP Telnet
  - Uses 3270 (VTAM LU2) interface to IMS
  - May use VTAM Generic Resources for IMS
- TCP/IP sockets to IMS Connect
  - IMS Connect provides TCP/IP support for IMS TM
    - Communicates with client using TCP/IP sockets
    - Communicates with IMS using OTMA interface
  - Requires TCP/IP based distribution of connections

Many installations access their IMS systems using TCP/IP. This includes connections from the web. Web servers can use many different ways of connecting to IMS. The most typical are:

1) APPC. If the web server sends requests to the host MVS using APPC protocols, then the connections to IMS can be distributed using APPC/MVS support for VTAM Generic Resources.

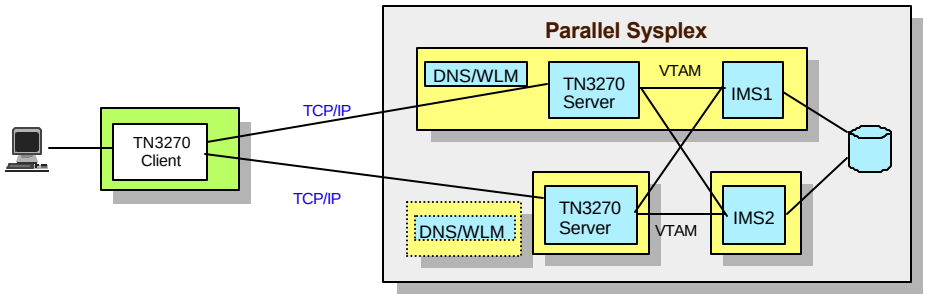
2) TCP/IP Telnet. TN3270 allows 3270 users to use TCP/IP protocols. The end user is a TN3270 client. The TN3270 client communicates with a TN3270 server using TCP/IP. The TN3270 server uses LU2 (3270) protocols to communicate with IMS via VTAM. VGR can be used with TN3270 servers to provide connection balancing.

3) TCP/IP sockets. If the web server uses sockets, it may communicate with IMS Connect which, in turn, communicates with IMS. IMS Connect executes in its own address space. It communicates with its client, in this case the web server, using TCP/IP socket protocols. It communicates with IMS using IMS's OTMA interface. OTMA (Open Transaction Manager Access) uses MVS XCF (Cross System Coupling Facility). XCF allows programs running in different address spaces, possibly on different MVSs in the Parallel Sysplex, to send and receive messages from each other. The distribution of connections from web servers to IMS Connect must be done with TCP/IP capabilities. We will see some of these possibilities..

DNS/WLM with TN3270

DNS/WLM provides "generic name" for TN3270 servers

- May be on one or more systems in the Sysplex
- Works well with long-lasting connections
  - Expensive (CPU-wise) for short-lasting connections
- VTAM Generic Resources may be used between TN3270 server and IMS
  - TN3270 server and IMS may be on different systems in the sysplex



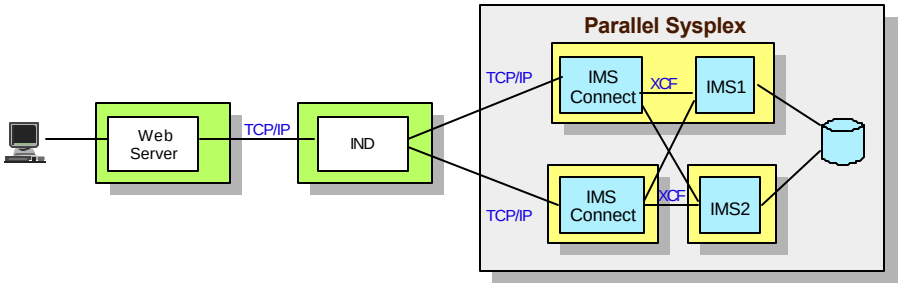
This is an illustration of Telnet 3270 use. The diagram shows four systems. On the upper one, there is a DNS/WLM, a TN3270 Server, and an IMS. The other three systems in the Parallel Sysplex have second copies of these.

DNS/WLM (Domain Name Server/Workload Manager) can be used in conjunction with TN3270 to distribute the connections requests across multiple TN3270 servers in the Parallel Sysplex. The TN3270 client request goes to one DNS/WLM which then uses the WLM to decide which TN3270 server should get the connection request. Once the DNS/WLM chooses a TN3270 Server, it is no longer involved. That is, communications goes directly between the TN3270 Client and the TN3270 server. The TN3270 server can then use VTAM Generic Resources to distribute sessions across the IMS members. VGR will always use a "local" IMS if one is available. A local IMS is one that is using the VTAM which the TN3270 server is using. On the other hand, if the TN3270 server is on an MVS image without an IMS, VGR can send the logon request to an available IMS on any MVS. The second DNS/WLM in the diagram is a backup for the first (in case the first fails).

While this configuration provides good connection balancing, it is fairly expensive (CPU-wise) to establish and terminate the connection. So, this is not a good configuration for connections which are short term.

IND with TCP/IP Web Connections

- **IND provides a "generic IP address" for IMS Connects**
  - IMS Connects may be on different systems
  - Efficient for short-lasting connections,
    - Requires separate hardware box (e.g. 2216 router)
  - IND function included in WebSphere Edge Server



The Interactive Network Dispatcher (IND) can be used to distribute connection requests from a web server to one of several IMS Connect address spaces in the parallel sysplex. IND is much more efficient than DNS/WLM at handling connection requests, but requires a separate hardware box (such as a 2216 router) and so can be more expensive. IMS Connect then sends work to one of several IMSs using XCF services and IMS OTMA.

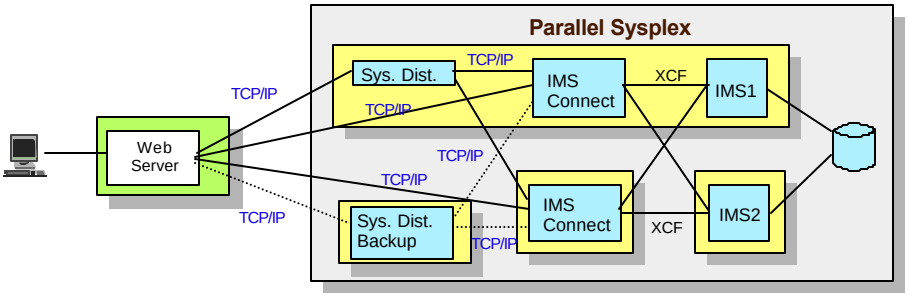
Note that both IMSs can be reached through either IMS Connect. Exit routines in IMS Connect may be used to choose to which IMS the request will be sent. These routines may have available to them information about which IMSs are currently active.

The network dispatching function of IND is included in WebSphere Edge Server. There are WebSphere Edge Servers for LINUX, AIX, Windows, and Sun Solaris.

Sysplex Distributor with Web

Sysplex Distributor was introduced in OS/390 V2R10

- Replacement for IND
- Does not require extra hardware box
- Sysplex Distributor runs in OS/390 or z/OS
  - Distributes connection requests across multiple servers (IMS Connects)
  - Works well with both short and long-lasting connections
  - Backup allows connections to survive an outage of the Sysplex Distributor without interruption



A better product for both long and short connections is the Sysplex Distributor. It is software that runs on the host system and distributes sockets across multiple target systems. In the case illustrated here, there are multiple instances of IMS Connect. Like DNS/WLM, a backup Sysplex Distributor can be running on another MVS in the sysplex.

Inputs go through the Sysplex Distributor. Responses do not go through the Sysplex Distributor. In the illustration they go directly from IMS Connect to the Web Server.

IMS Transaction Manager

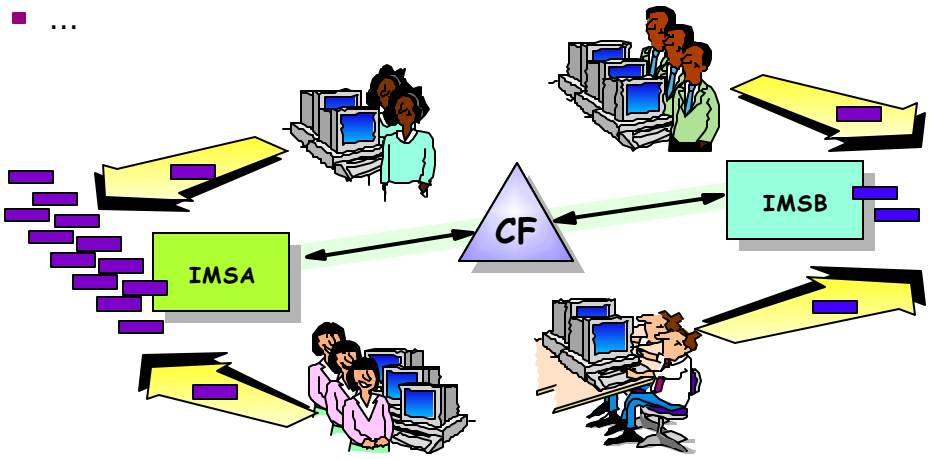
Step 3: Distributing Transactions

The third step in using Parallel Sysplex with IMS Transaction Manager is the distribution of transactions. This involves processing a transaction on a system other than the one which initially received the input message.

Distributing the Workload

Balancing the connections may not balance the workload

- Batch work could overload some systems
- Peaks may occur differently from different connections
- ...



The techniques we use to distribute connections from users across our IMS systems may not balance our workload. Several things can cause this. First, a large batch workload on one system may overload it. Users who are already connected to that system remain connected to it. Their response times could be affected by this overload. Second, the volumes of inputs from connected users may vary. This could result in peak loads on one system while another system has a lull in inputs. Other factors could also cause unbalanced workloads across the systems.

We will now look at techniques to address this imbalance. They involve distributing transactions to other IMS TM systems.

Distributing Transactions

Distributing transactions

- Multiple Systems Coupling (MSC)
  - IMS sends transactions to partner system for processing
- Shared Queues
  - IMS systems share one set of message queues
  - Any IMS can process messages on the queues
- Neither depend on where end user is logged on

When we distribute transactions, an input transaction may be routed from one IMS to another for processing. There are two methods of doing this in IMS.

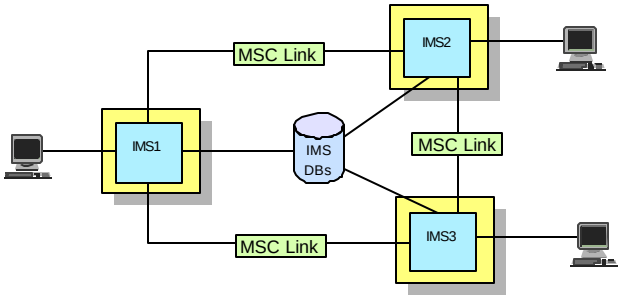
1) Multiple Systems Coupling (MSC). With MSC, multiple IMS systems are connected by communication links. An IMS system may send a message across one of these links to another IMS. The receiving system processes the transaction and sends its reply to the original system. The original system sends the reply to the user.
2) IMS Shared Message Queues (SMQ). In this implementation, the IMS systems share one set of message queues. They are stored in list structures in coupling facilities. Since the queues are available to all the IMS systems, any IMS system may process a transaction. Those with more processing capacity tend to process more transactions. Shared message queues were introduced in IMS Version 6.

With either of these implementations, a user may be connected to any system and have an input message processed on another IMS system. There are some limitations for APPC and OTMA connections. In some cases, input messages from these connections must be processed of the system where the connection exists.

Routing Messages with MSC

Multiple Systems Coupling (MSC)

- MSC routes transactions between IMSs
  - Distribute workload across multiple IMSs
- Not dynamic
  - Local/remote transactions defined in IMSGEN
    - Transaction destination predetermined
  - Difficult to add IMSs to environment
  - No workload balancing capability
  - IMS or link failure may cause unavailability



As this illustration shows, MSC is comprised of VTAM sessions between multiple IMS systems. When an IMS transaction is received by any IMS, its definitions determine where that transaction is to execute. It may be processed locally, that is, on the IMS system which receives it. On the other hand, it may be processed remotely, that is, on another system. If it is defined to run remotely, the IMS system which receives the message sends it to the remote system.

MSC can be used to distribute transactions across multiple IMSs. The definitions are static. An IMS makes its decision about whether or not to process a transaction based on the transaction code. This is not dynamic. Decisions are not based on workloads. It is relatively difficult to add a new IMS system to the complex. A new IMS system requires changes in the definitions for the other existing IMS systems.

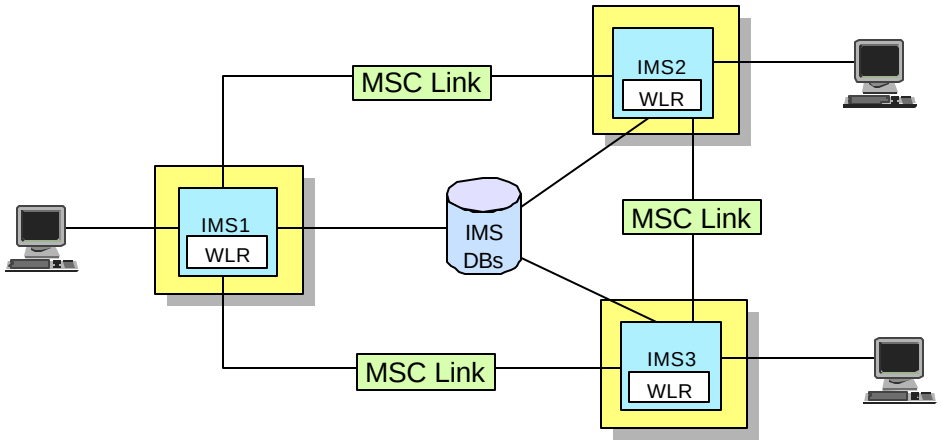
MSC definitions may be used to distribute the workload, but they do not balance the workload.

A link failure or an IMS failure may mean that a transaction cannot be processed until the failure is corrected.

IMS/ESA Workload Router

IMS/ESA Workload Router

- Uses MSC to route IMS TM transactions
  - Set of MSC exit routines
- May be used to distribute workload
  - Some balancing capability

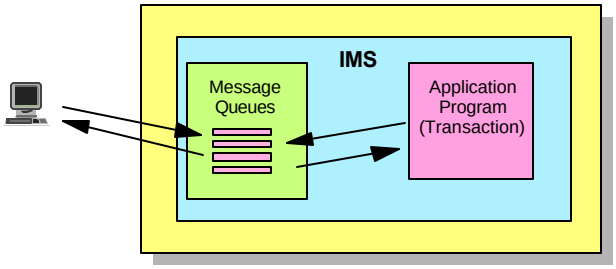


MSC users may include MSC exit routines to override the definitions of where transactions are processed. This adds some dynamic capabilities to MSC routing. The IMS/ESA Workload Router product provides a set of these exit routines.

The Workload Router (WLR) uses MSC Directed Routing to distribute transactions across multiple IMSs without regard to how they have been defined. For example, if TRANA is received at IMS1, it may execute on IMS1, IMS2, or IMS3. The WLR can be directed to process a certain percentage of transactions locally and to send others to remote IMSs. This provides some workload balancing capability.

IMS Message Queues

- **IMS places messages in queues**
  - Messages may be received from terminals or programs (transactions)
  - Messages may be sent to terminals or programs (transactions)
- **Conventional queues reside in one IMS system**
  - Not available to other IMSs



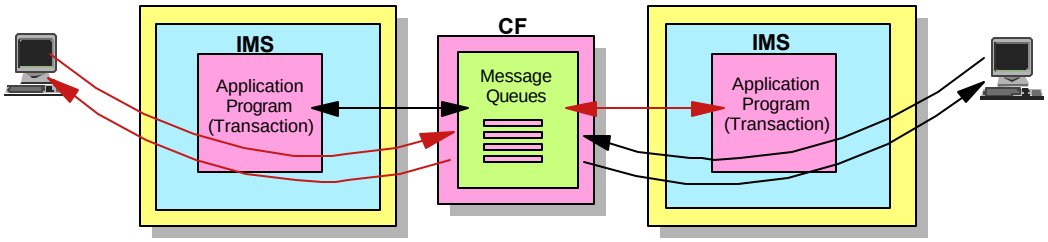
IMS TM is a queue driven system. Transaction input messages may be received from terminals or programs. Output messages may be sent to terminals or programs. All of these messages are placed in message queues.

In conventional (non-shared queues) systems, each IMS has its own set of queues. These queues are accessible only by the IMS system which owns them. When MSC is used, one IMS sends a message from its queues to another IMS system which places the message in its queues. In any case, a message may be processed only by the IMS system on whose queues it resides.

Shared Message Queues

IMS V6 introduced shared message queues

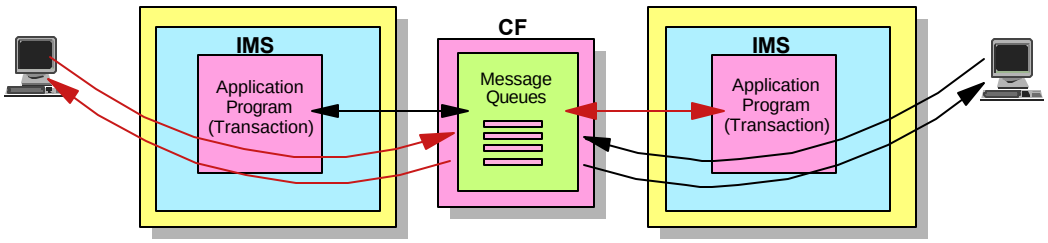
- Message queues moved to coupling facility list structures
  - Single set of queues
- Multiple IMS systems share one set of queues
  - IMS systems place messages in common queues
  - IMS systems retrieve messages from common queues



With shared queues, the message queues are moved to list structures in coupling facilities where they are available to any IMS in the shared queues group. A terminal is connected to one IMS system. Input messages from the terminal are placed in the shared queues. They are accessible from any IMS using those queues. This means that another IMS system, not the one to which the terminal is connected, may process the input message.

Shared Message Queues ...

- **All messages are placed on the shared queues**
  - Transactions and responses
- **IMS subsystem registers interest in some queues**
  - Queues for transactions it can process
  - Queues for terminals which are logged on to that IMS
- **Each IMS with interest is notified when a queue becomes non-empty (work to be done)**
  - IMS may ask for message from non-empty queue
- **Only one IMS receives a message**

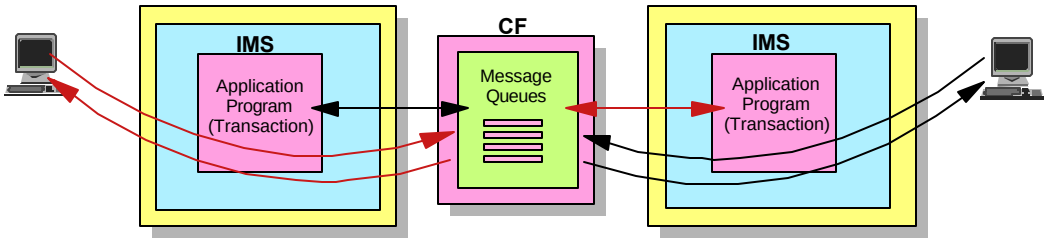


All messages (input and output) go into the shared queues. IMS subsystems register interest in specific queues, such as the queue for transaction TRANA or for terminal TERMX. IMS systems register interest for queues it can process. This includes the queues for the terminals connected to it and the transactions which are defined to it. When there is work on a registered queue, the IMS systems which have registered interest in the queue are notified. When an IMS has the resources available to process the transaction, such as an available dependent region, it attempts to read a message from the shared queue. If it receives a message, it processes it and puts the response back on a shared queue. Multiple IMSs may attempt to retrieve messages from a queue, but only one will receive any individual message. Terminal output messages are retrieved from the queue by the IMS to which the terminal is connected. This IMS sends the output message to the terminal.

Shared Message Queues ...

Balancing the workload with shared message queues

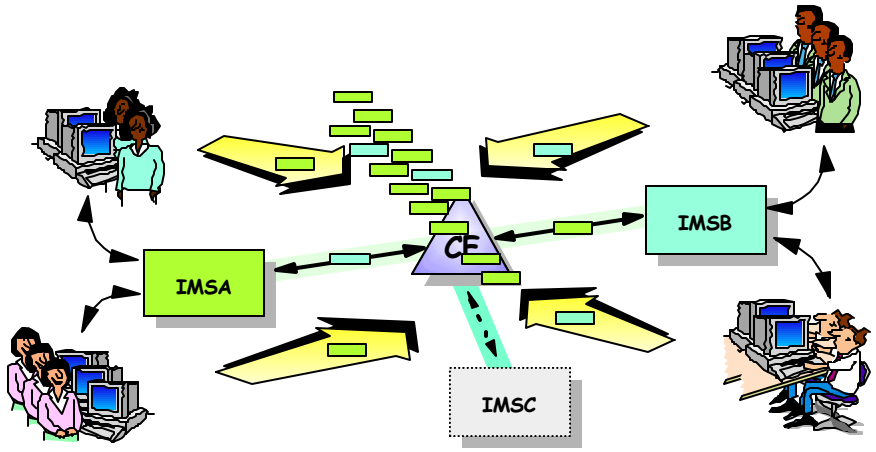
- All IMS systems have access to transaction messages
- IMS systems with available resources ask for work
- IMS systems with the most available resources will ask for work more frequently



Because any IMS can process a shared queues message, only those IMS systems with available resources will ask for and process the transaction. This tends to distribute the workload to the systems best able to handle it. Those systems with the most free resources will ask for work most frequently.

Shared Message Queues ...

- **Shared queues dynamically balance the workload!**
  - Transaction entered on IMSA may be processed by IMSB
  - True workload balancing
- **May be used in combination with connection balancing**

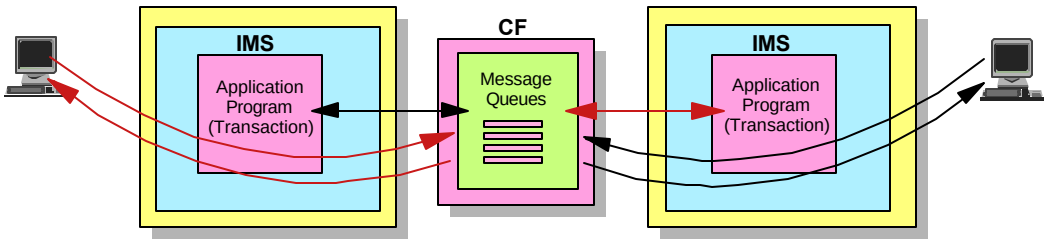


So, the application workload is balanced dynamically. If there is any processing capacity available anywhere in the shared queues group, queued transactions will be scheduled and processed. The user is not forced to wait because a single IMS is overloaded.

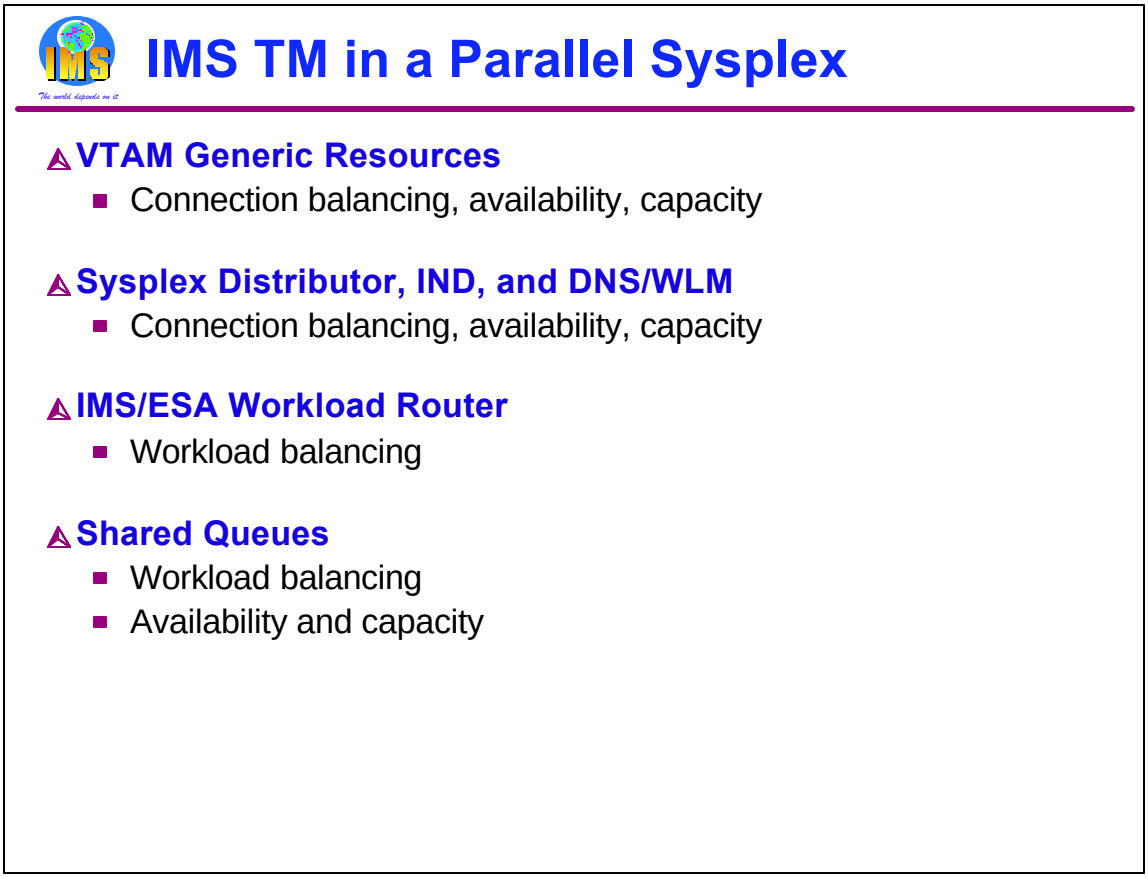
Note that shared queues can be used in conjunction with connection balancing provided by VTAM Generic Resources or one of the techniques used for TCP/IP.

Shared Message Queues ...

- **Availability**
  - Messages are available from any IMS system
- **Capacity**
  - New systems may be added without changes to existing systems or user logon procedures
  - Transaction workload is automatically shared
- **Workload balancing**
  - Work is dynamically distributed to systems with available capacity



Shared queues address all three of our concerns. Increased availability is provided by allowing any IMS to process a message, even if an IMS fails. Capacity is easily added since any number of IMS systems may be used to handle the workload. New systems may be added dynamically. Workload balancing is dynamic since IMSs with capacity will ask for more work by trying to process messages on the shared queues.



So, in a parallel sysplex, the IMS user can take advantage of multiple capabilities.

VTAM Generic Resources provides connection balancing for SNA sessions. It improves availability for users connected via VTAM and makes it easy to add new systems without changing user logon procedures.

The various TCP/IP distributors provides connection balancing for users of TCP/IP. They provide improved availability for these users and make it easy to add capacity for systems with these users.

The IMS Workload Router provides workload balancing capabilities for users of conventional queuing.

Shared queues provides dynamic workload balancing by allowing an IMS system with available capacity to process any transaction in the shared queues group. Shared queues provides availability benefits by allowing any surviving IMS to process transactions when another IMS fails. Capacity is easily added since no modifications have to be made to the previously existing IMS systems.

Rapid Network Reconnect

This section explains Rapid Network Reconnect which was added by IMS Version 7. Rapid Network Reconnect is optional. We will see that it may provide great benefits for some environments. On the other hand, it is not appropriate for all environments.

Rapid Network Reconnect (RNR)

Rapid Network Reconnect (RNR) implements support for VTAM Persistent Sessions

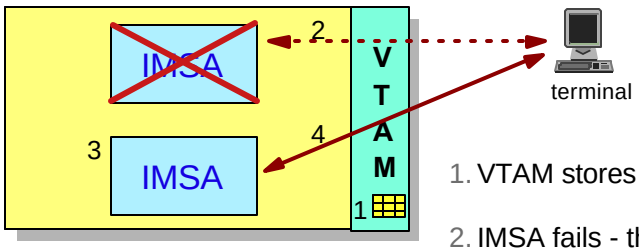
- Added in IMS Version 7
- Eliminates session cleanup/restart following an outage
  - Multinode Persistent Sessions (MNPS) support all host outages
    - IMS, VTAM, OS/390, and processor
  - Single Node Persistent Sessions (SNPS) support only IMS outages
    - IMS abends
- Sessions survive an outage
  - Users remain logged on while awaiting IMS restart
  - Session bind and unbind traffic does not flow through the network
  - Sign-on may be required

IMS V7 delivered support for VTAM persistent sessions. In IMS this is called Rapid Network Reconnect (RNR). This eliminates session clean up and restart when a host failure occurs. There are two kinds on persistent session support. Multinode persistent sessions (MNPS) provide support for all types of host failures. These include failures of IMS, VTAM, OS/390 (or z/OS) or the processor. Single node persistent sessions (SNPS) provide support only for IMS failures. With SNPS the VTAM instance must survive.

With persistent sessions, end users do not lose their sessions for the supported failures. In fact, they remain logged on. Even though their IMS system fails, their sessions are not terminated. This means that the unbind traffic does not flow through the network when the failure occurs. Secondly, when their IMS system is restarted, their sessions do not have to be reestablished and the bind traffic does not flow. For LU types which typically have human users, such as SLUTYPE2 and SLUTYPE1 (CONSOLE), sign-on is required. For LU types which typically are programmable, such as SLUTYPEP, or do not have direct human users, such as LU1 (PRINTER1), sign-on is not required.

Single Node Persistent Sessions

Single Node Persistent Session scenario

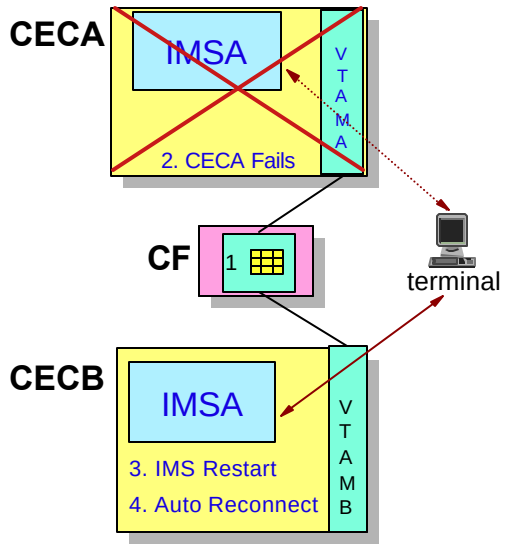


1. VTAM stores session data in data space
2. IMSA fails - the session to terminal is pending recovery
3. IMSA restarts
4. Auto reconnect of terminal to restarted IMS
   - **Terminal remains logged on**

When using RNR with SNPS, only outages due to IMS abends are supported. The VTAM used by this IMS must survive. When a session is established, session data is stored in an MVS data space associated with the VTAM address space. If the IMS system abends but VTAM survives, the session survives and the session data remains. When the IMS is restarted, the users sessions are given to the restarted IMS. These users have remained logged on even though their IMS system has failed.

Multinode Persistent Sessions

Multinode Persistent Session scenario



1. VTAM stores session data in the CF
2. CECA fails
   - Another VTAM in the Parallel sysplex detects the error
3. IMSA is restarted on CECB
4. The sessions are restarted using information saved in the CF structure
   - **Terminal remains logged on**

With MNPS, the session data is stored in a CF structure where it is available to other systems in the sysplex. All types of failures are supported with MNPS. As with SNPS support, when IMS restarted, the users are automatically reconnected in a "logged on" state.

When using RNR with MNPS, all outages of the IMS, VTAM, or processor are supported. When a session is established, session data is stored in a coupling facility structure. In the illustration the processor fails. The session data is not lost. The session survives. The IMS may be restarted on another system in the Parallel Sysplex. When IMS is restarted, the users sessions are given to the restarted IMS. These users have remained logged on even though their IMS system has failed.

RNR Benefits

- **Session termination and establishment traffic eliminated**
  - Session information is maintained
- **Terminal service to same IMS is reestablished more quickly**
  - Valuable to users without cloned systems
- **Value of RNR depends on how quickly IMS is restarted**

Persistent session support for APPC is provided by APPC/MVS

- Sessions are persistent, conversations are not

The benefit of RNR is the maintenance of the sessions when IMS fails. Most of this benefit is the elimination of the time required to terminate and reestablish the sessions. This eliminates the bind and unbind traffic which would otherwise flow through the network. This traffic can be time consuming. Service to the end users is reestablished more quickly. Of course, the IMS system has to be restarted. When using RNR, the end user does not have the option of logging on to another IMS in the Parallel Sysplex. The value of RNR depends on how quickly IMS is restarted. If the restart is slow, there is not much benefit. If the restart is quick, the benefit can be substantial. On the other hand, if another system with the same capabilities is available, the users would get quicker restoration of service by logging onto it. This means that RNR probably will not be used for IMS systems with clones.

Persistent session support for IMS users of APPC (LU6.2) is provided by APPC/MVS, not IMS. With APPC the sessions are persistent, but the conversations are not.

Parallel Sysplex Failure Recovery

Failure Recovery

The final section of this presentation covers recoveries from failures. Parallel Sysplex adds more components to a system. These includes clones of systems and subsystems, as well as, new components such as Coupling Facilities and CF links. Even though we may have another component available to do our work when one component fails, we want to restore our sysplex to full robustness as soon as possible. We will see how recoveries from most failures in a Parallel Sysplex may be automated.

Failure Recovery

Parallel Sysplex takes advantage of multiple servers

- Multiple subsystems (IMS, DB2, or CICS)
  - A failure of one may cause its workload to be moved to others
    - VTAM Generic Resources, Sysplex Distributor, Shared Queues
- Multiple S/390 processors
  - A failure of one may cause its subsystems to be moved to others
    - Automatic Restart Management (ARM)
- Multiple Coupling Facilities
  - A failure of one may cause its structures to be moved to others
    - Structure Rebuild
  - Implementers may create multiple copies of a structure
    - Eliminates need to rebuild on a failure

The obvious thing to note in a Parallel Sysplex is that we have multiple copies of our servers. When one fails, another is available to do its work. This applies to subsystems, processors, and Coupling Facilities.

If IMS fails, other IMS instances are available. We can use our routing and balancing capabilities to distribute the work to the surviving IMS subsystems.

If a processor or LPAR fails, we will see that Automatic Restart Management (ARM) can be used to restart failed subsystems on surviving processors or LPARs.

If a Coupling Facility fails, we have two ways of surviving the loss. In some cases, we rebuild its structures on another CF. In others, implementers use multiple copies of the structures.

MVS, Processor, and IMS Failures

When system or IMS fails we need to restart IMS

- Resolve in-flight and in-doubt work
- Release locks and DBRC authorizations

Automatic Restart Management (ARM)

- ARM can restart IMS (DB2, CICS, etc.) on ABENDs and system failures
- Restarts for system failures move work to another system

When IMS fails, we need to restart it as quickly as possible. Even though other IMS subsystems may be available to do work, the failed IMS may have in-flight or in-doubt work that needs to be resolved. This resolution releases locks on database resources and releases DBRC authorizations. This allows new work to have access to all of the data.

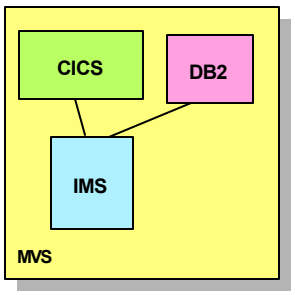
Automatic Restart Management (ARM) may be used to provide rapid restarts of IMS. ARM is a sysplex capability that allows an automatic restart of subsystems like IMS, DB2, CICS, and IRLM. If the subsystem abends, the restart is on the same MVS instance (LPAR). If the MVS (LPAR) fails, the restart is on another MVS in the sysplex.

ARM

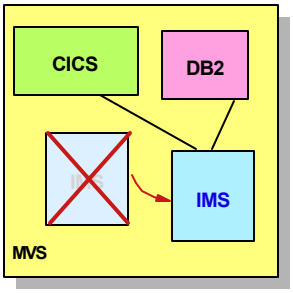
IMS Abend

- ARM restarts IMS on same MVS

Before Abend



After Abend and ARM Restart

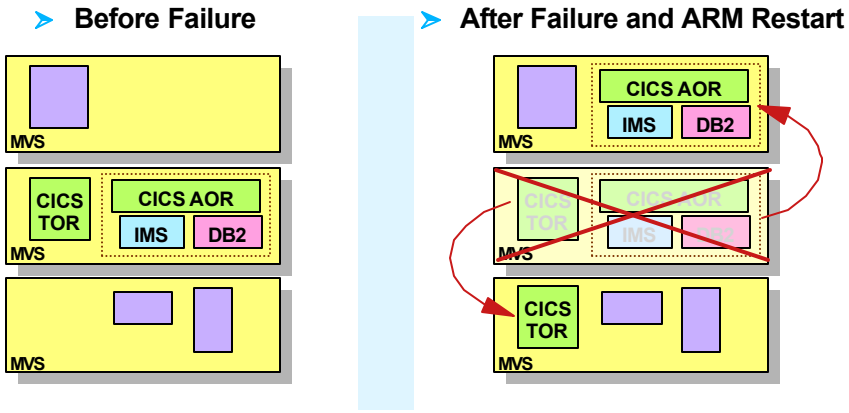


This example illustrates the actions of ARM when IMS abends. IMS is restarted on the same MVS system. This IMS was providing DBCTL services to a CICS and was using a DB2 subsystem. Clearly, we need to restart IMS on the same MVS so that services between these subsystems may be restored. For example, in-doubt threads must be resolved.

ARM ...

MVS or system failure

- ARM restarts IMS on another MVS in the sysplex
- Subsystems may be grouped for restart on same MVS



In the case of an MVS or processor failure, IMS will be restarted on another candidate MVS. The MVS is chosen according to a user defined ARM policy. Subsystems which must remain together may be restarted as a "group" on the same MVS. In this example, an IMS subsystem is using DB2 for database services. A CICS AOR (Application Owning Region) is using the same DB2 and the IMS for database services. When the MVS system fails, the IMS, the DB2, and the CICS AOR must be moved together. On the other hand, the CICS TOR (Terminal Owning Region) may be restarted on another MVS in the sysplex since CICS AORs and TORs can communicate across MVSs.

Using ARM with IMS

- **ARM must be active in the sysplex**
  - ARM policy defined and started
- **IMS must register with ARM**
  - ARMRST=Y execution parameter (default)
- **IMS has ARM support for ...**
  - IMS control region
  - CQS (shared queues)
  - FDBR
  - IRLM
- **ARMWRAP is available for IMS Connect**

For ARM to restart subsystems, it must be active in the sysplex. ARM is controlled by a policy which the user defines. The policy is stored in an ARM Couple Data Set. The policy is used to group subsystems for restart together. It also controls whether or not a subsystem will be restarted. For example, an installation may not want to restart test subsystems.

ARM only restarts subsystems which register with ARM. This is done when they initialize. IMS has a parameter (ARMRST) which controls whether or not IMS registers with ARM. ARMRST=Y is the default.

IMS has full ARM support. ARM can be used to restart IMS control regions, Common Queue Server regions, Fast Database Recovery regions, and IRLMs. ARM does not directly restart IMS dependent regions. These are typically started by automation when the control region is started.

ARMWRAP is a program which registers an address space for ARM restarts. In is used for a step in a job. If the following step fails, ARM will restart the job. IMS Connect does not register with ARM. ARMWRAP may be used to get ARM support for IMS Connect.

CF and CF Link Failures

Loss of Coupling Facility

- Structures in failed CF are lost
  - May be rebuilt on another CF
  - May have duplicate structure on another CF

Loss of CF link

- Access to structures is lost
  - May be treated like loss of CF
  - Structures may be rebuilt or duplicate structure used

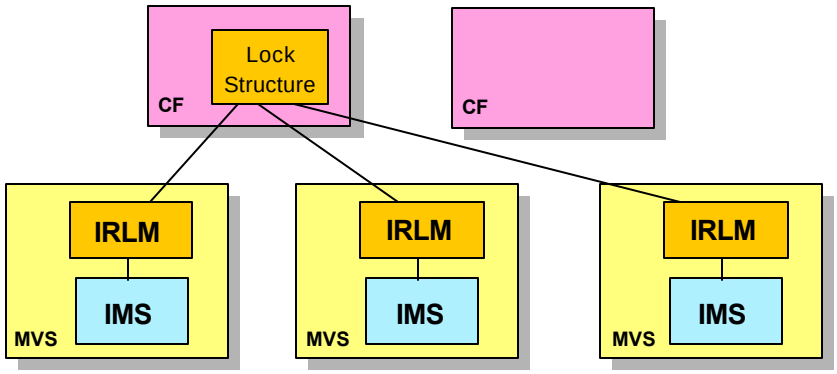
Much of the Parallel Sysplex support is provided through the use of coupling facility structures. If a CF is lost, it is important to have access to structures elsewhere. In come cases this is done be rebuilding the structures. In others, this is done by having duplicate structures on another CF.

If a CF survives, but we lose all of the links from a processor to the CF, we need to resolve the problem. This may be treated like the loss of the CF itself. That is, we may either rebuild its structures on CFs which have connectivity to the processors which require it or we may use duplicate structures.

The following pages show examples of these situations.

Structure Rebuild

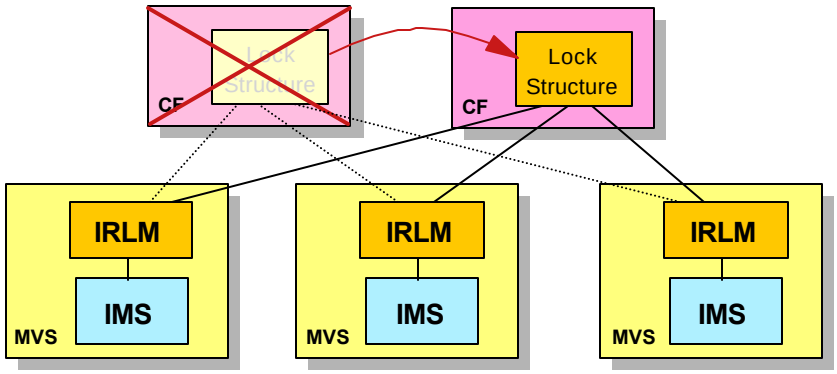
- **Structure rebuild is used for**
  - IRLM lock, OSAM, VSAM, and Shared Queues structures
- **IRLM lock structure example:**



Some structures can be rebuilt automatically when either a CF failure or a CF link failure occurs. These include IRLM lock structures, OSAM and VSAM cache structures, and IMS shared queues structures. The example will use a IRLM lock structure.

Structure Rebuild

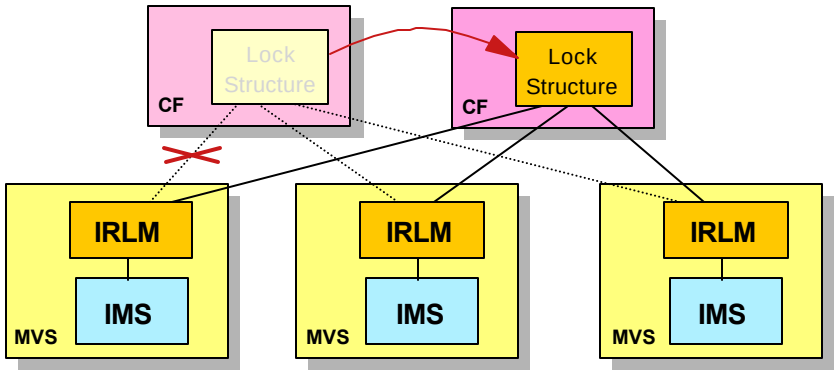
- Structure rebuild is used for loss of CF
- IRLM lock structure example:



In this example, we lose the CF on which the lock structure resides. When the CF fails, the system automatically recognizes the loss and rebuilds the lock structure on another CF. Each IRLM retains the information necessary to restore its lock information in the structure. The IRLMs together rebuild the lock structure on another CF. Data sharing is resumed. Similar rebuild and recovery occurs for OSAM, VSAM, and shared queue structures.

Structure Rebuild

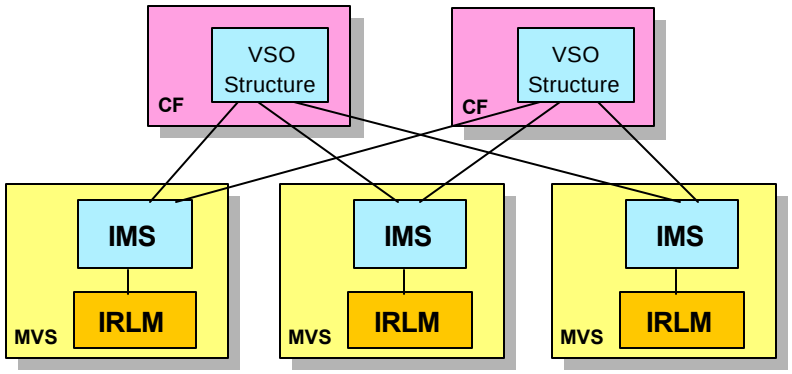
- Structure rebuild is used for ***loss of connectivity to CF***
- IRLM lock structure example:



In this example, the CF does not fail. Instead, the connectivity between one of the processors and the CF fails. This case is treated the same as the loss of the CF. That is, the system automatically rebuilds the structure on another CF. All processors have connectivity to this CF. This means that data sharing can continue. Similar rebuild and recovery occurs for OSAM, VSAM, and shared queue structures.

Structure Duplexing

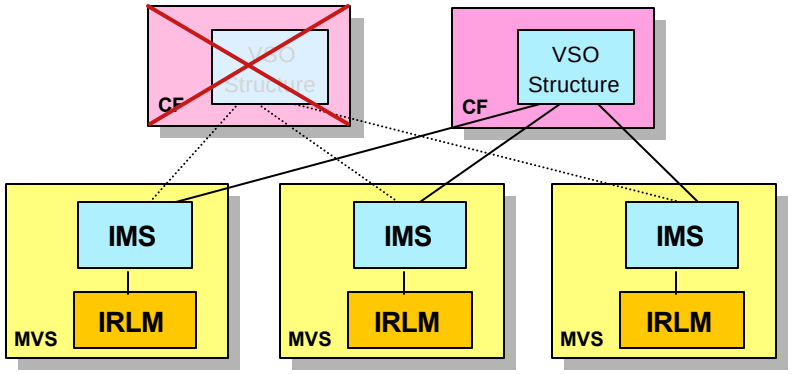
- **User-manged structure duplexing is available for**
  - DEDB VSO structures
- **System-managed structure duplexing is available for**
  - DEDB VSO, IRLM lock, and Shared Queues structures
- **DEDB VSO structure example:**



Fast Path shared VSO does not rebuild its cache structures. Instead, it relies on a duplicate copy to provide failure survival. The duplicate copy may be created in either of two ways. First, Fast Path may build two structures. This is user-manged duplexing. Second, users of z/OS V1.2 with the appropriate hardware prerequisites may have the system build duplexed structures. This is system-manged duplexing. System-manged duplexing is also available for IRLM lock structures and Shared Queues structures.

Structure Duplexing

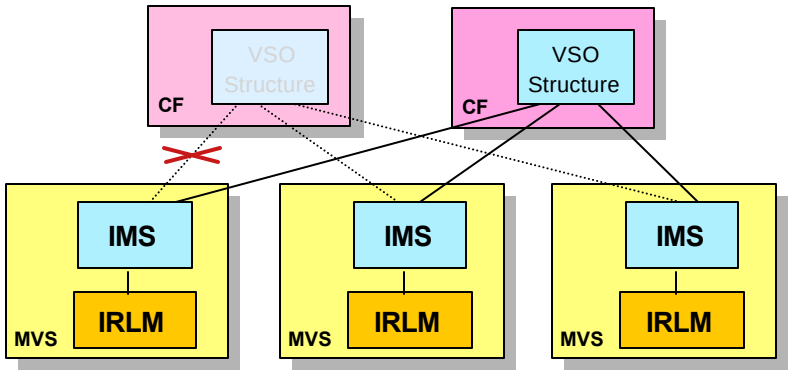
- Structure duplexing is used for ***loss of CF***
- DEDB VSO structure example:



If a CF is lost, then a duplicate structure on another CF is used. Stated another way, we simply discontinue using the structure we lost. With system-managed duplexing a duplicate is immediate built if another CF is available. If not, a duplicate is built when another CF becomes available.

Structure Duplexing

- Structure duplexing is used for ***loss of connectivity to CF***
- DEDB VSO structure example ...

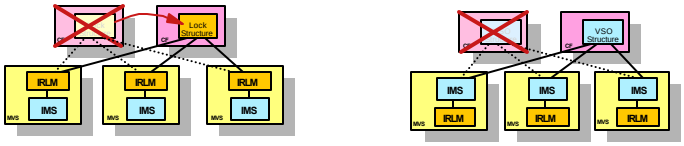


Similarly, if connectivity to a CF is lost, then the use of its structure is discontinued. The duplicate structure on another CF is used instead.

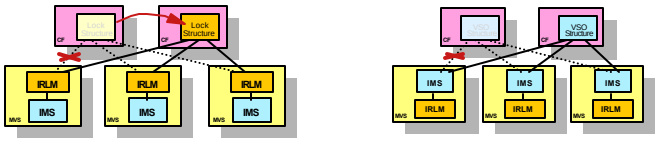
Parallel Sysplex Failure Recovery

- ARM to restart IMS, CICS, DB2, etc.

- Structure rebuild and/or duplexing to handle CF failures



- Structure rebuild and/or duplexing to handle CF link failures



So, failure recovery from various types of failures in a Parallel Sysplex can be automatic.

ARM may be used to restart failed subsystems. These failures may be abends or failures of the processors on which the subsystems run.

If a CF is lost, either rebuild or duplication is used to recover.

If connectivity to a CF from a processor is lost, either rebuild or duplication is used to recover.

IMS in a Parallel Sysplex

High availability

- Data sharing, shared queues, generic resources, Sysplex Distributor, IND, DNS/WLM, RNR (persistent sessions), ARM, structure rebuild, duplex structures

Increased capacity

- Data sharing, shared queues, generic resources, Sysplex Distributor, IND, DNS/WLM, IMSGROUP

Workload balancing

- Generic resources, Sysplex Distributor, IND, DNS/WLM, Workload Router, shared queues, IMSGROUP

In this presentation we have seen how the various capabilities available with IMS in a Parallel Sysplex can be used to improve availability, increase capacity, and balance workloads. These lists summarize which capabilities may be used to provide the different benefits.

More Information

Redbooks

- ***IMS/ESA V6 Parallel Sysplex Migration Planning Guide for IMS TM and DBCTL***, SG24-5461
- ***IMS/ESA Data Sharing in a Parallel Sysplex***, SG24-4831
- ***IMS/ESA Shared Queues: A Planning Guide***, SG24-5257
- ***IMS/ESA Version 6 Shared Queues***, SG24-5088

The Web

- **www.ibm.com/ims**
  - Follow link to Library
    - http://www.ibm.com/software/data/ims/library.html
  - Follow link to "Presentations/Papers"
    - http://www.ibm.com/software/data/ims/shelf/presentations/index.html
- **www.ibm.com/support/techdocs**
  - Search for IMS in "Presentations & Tools"
    - Document PRS160, *IMS V7 Presentation*
    - Document PRS174, *Automatic Restart Management (ARM) with IMS*

There is more information available on IMS in a Parallel Sysplex. Some of the sources are shown here. The information in the IMS/ESA V6 Redbooks is also applicable to later IMS releases.

These sources of information are in addition to the standard IMS, OS/390, and z/OS product publications.

Acronyms and Terms

ARM - Automatic Restart Management
AOR - Application Owning Region (CICS)
BLDS - Block Level Data Sharing
BMP - Batch Message Program
CEC - Central Electronic Complex
CF - Coupling Facility
CQS - Common Queue Server (Shared Queues)
DEDB - Data Entry Database (Fast Path)
DNS/WLM - Domain Name Server/Workload Manager
EMH - Expedited Message Handler (Fast Path)
FDBR - Fast Database Recovery
IFP - Interactive Fast Path Region
IND - Interactive Network Dispatcher
ITOC - IMS TCP/IP OTMA Connection
MSDB - Main Storage Database (Fast Path)
MSC - Multiple Systems Coupling
OTMA - Open Transaction Manager Access
SDEP - Sequential Dependent of DEDB
TOR - Terminal Owning Region (CICS)
VSO - Virtual Storage Option for DEDB
WLM - Workload Manager
WLR - Workload Router
XCF - Cross System Coupling Facility

This page lists some of the acronyms and terms used in this presentation.