



Software Group | Enterprise Networking and Transformation Solutions (ENTS)

z/OS IP Load Balancing Advisor (LBA) for External Load Balancers: Implementation and Recovery Scenarios

© 2005 IBM Corporation

z/OS Load Balancing Advisor (LBA) - agenda

- How to implement the z/OS Load Balancing Advisor (LBA)
- Recovery scenarios





How to implement the z/OS Load Balancing Advisor (LBA)

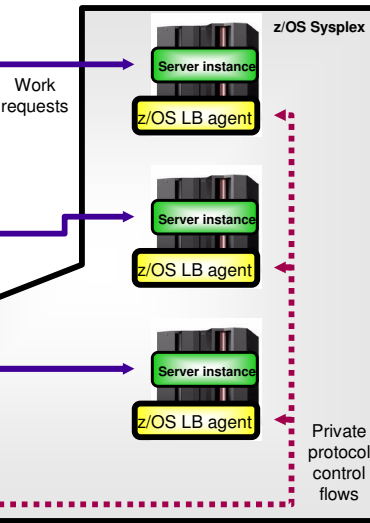
z/OS Load Balancing Advisor (LBA) to improve quality of load balancing decisions made by outboard load balancers

The Server Application State Protocol (SASP) control flows will provide relative weights per server instance (based on WLM weight, server availability, and server processing health taking such metrics as dropped connections, size of backlog queue, etc. into consideration).

SASP is also used by the new eWLM infrastructure.



Cisco CSM currently supports the SASP protocol.



z/OS workload balancing

- Support for clustered z/OS servers in a z/OS Sysplex
- Not focused on HTTP(S) only, will support all IP-based application workloads into a z/OS Sysplex
- Based on Sysplex-wide WLM policy
- Scope is a z/OS Sysplex

The z/OS Load Balancing Advisor technology is a new z/OS Communications Server technology that was made generally available in 4Q2004 via APARs PQ90032 (V1R4) and PQ96293 (V1R5/V1R6). It is fully integrated into z/OS V1R7.

Notes - z/OS Load Balancing Advisor

NOTES

- Load balancer (center of the diagram)
 - ⌋ Determines which server instance in the Sysplex should receive new workload using information from Advisor
 - ⌋ Workload arrives from clients (leftmost in diagram) and performs workload balancing to z/OS applications according to server availability, system capacity, and server ability to handle new workload
 - ⌋ CISCO support for SASP is available on their Content Switching Module (CSM)
 - Requires CSM level 4.1 (2.5)
- The z/OS Load Balancing Advisor solution consists of two z/OS applications
 - ⌋ Advisor (bottom-center of diagram)
 - One instance per Sysplex
 - Communicates with external load balancers using the SASP protocol
 - Provides external load balancers with information about z/OS resources
 - Collects information from agents using a private protocol
 - ⌋ Agent (rightmost three z/OS images in diagram)
 - One instance per MVS system
 - Collects information about z/OS systems and applications
 - Sends information to advisor for aggregation using the private protocol
- Note: Diagram does not show an agent on the system where the advisor is located, but this can be done and is required if applications on that system want to be targets of workload distribution.

Load balancer configuration

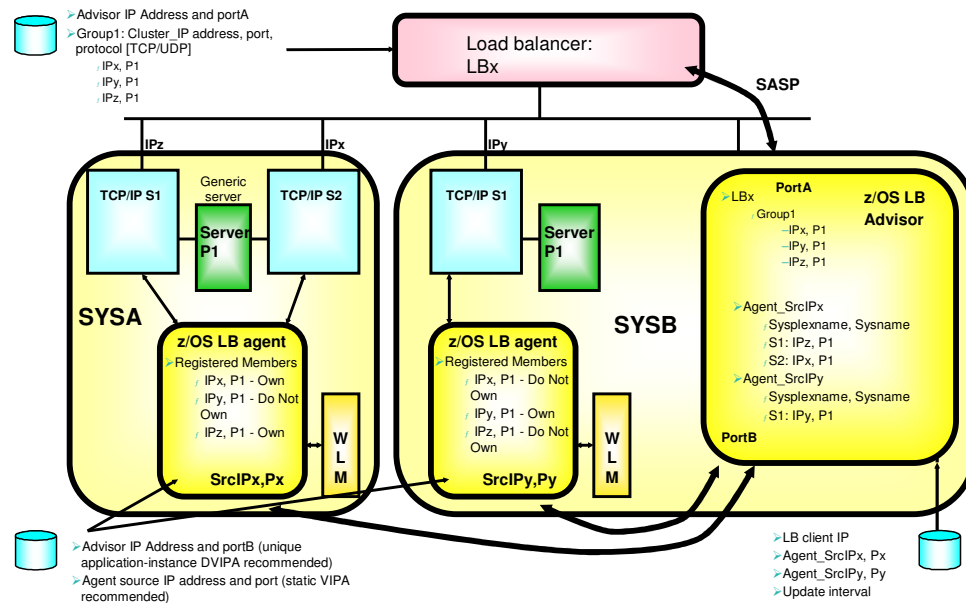
➤ Group definition rules/guidelines

- ⌈ Multiple groups may be defined representing different server clusters
- ⌈ All members of a group must belong to the same Sysplex
 - Therefore, members of the same group must all be managed by the z/OS Load Balancing Advisor, or EWLM. A group may not be managed by both.
- ⌈ May not contain mixtures of application members and system members
- ⌈ Clients connect to the cluster IP address of the group

➤ Member definition rules/guidelines

- ⌈ IP addresses should be VIPA addresses for availability reasons
- ⌈ Should NOT contain the following types of addresses
 - Distributed DVIPAs
 - "Deprecated" IPv6 addresses
 - Addresses that are not unique within the Sysplex
 - Addresses that would not be reachable from the Load Balancer, including
 - Loopback addresses
 - "Unavailable" IPv6 addresses

z/OS LB Advisor/Agent structure - overview



The weights

- The weights are composed of two main elements:

- **WLM weight**

- The WLM weight that we know from other WLM-based load balancing solutions, such as Sysplex Distributor
 - ✓ A numeric value between 0 and 64

- **Communications Server weight**

- This weight is calculated based on the availability of the actual server instances (are they up and ready to accept workload) and how well TCP/IP and the individual server instances process the workload that is sent to them?
 - ✓ Expressed as a numeric percentage value between 0 and 100
 - Purpose of calculations is to:
 - ✓ Prevent stalled server from being sent more work (accepting no new connections and new connections are being dropped due to backlog queue full condition)
 - ✓ Proactively react to server that is getting overloaded (accepting new connections, but size of backlog queue increases over time approaching the max backlog queue size)

- The final weight is calculated by combining the WLM and the CS weights into a single metric

- Final weight = WLM weight * CS weight / 100

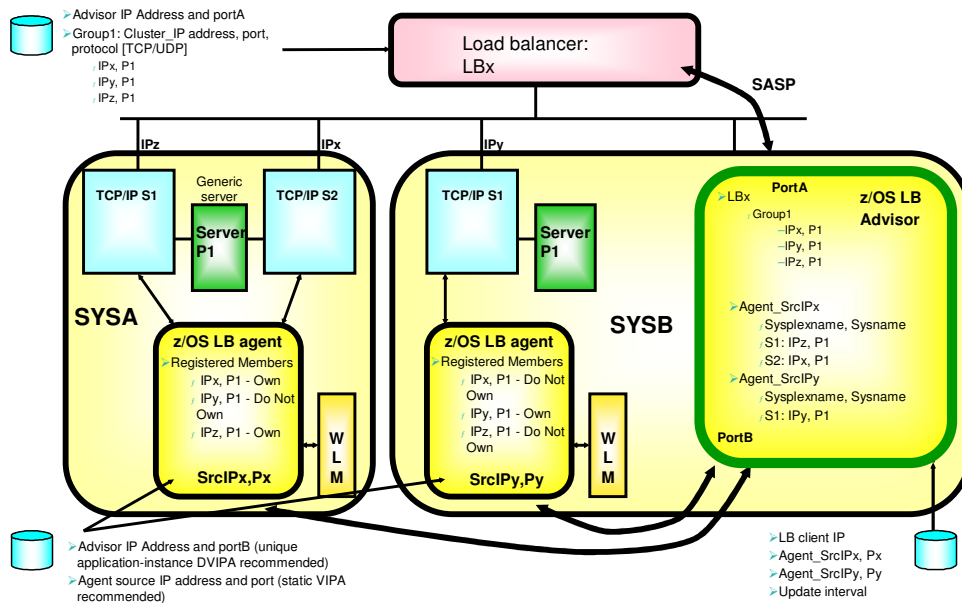
- Due to current external load balancer behavior when a weight of zero is returned for all members of a group, the z/OS LBA currently will never return a zero weight for all members in a group

- In the case that all members indeed do have a weight of zero, they will all be reported to the LB as having a weight of one

- Weights that are returned to the load balancer are normalized to values between 1 and 64

- If all server instances have the same final weight (example 32), then a 1 will be returned for all server instances

z/OS LB Advisor address space - configuration overview



z/OS Load Balancing Advisor

➤ **New, stand-alone application**

- Started via an MVS Started Task
- Accepts MVS operator commands for display and modification purposes

➤ **Executes on any system within the sysplex**

- Provides load balancing advice for any TCP/UDP server applications within the Sysplex
- Acts as a TCP server application supporting SASP (port 3860 by default, but can be customized)
- Supports multiple LBs concurrently
- Only one active instance allowed per Sysplex
- Does not require Sysplex Distributor to be configured

➤ **Communicates with local load balancing agents**

- Uses TCP connections, acts as TCP server (on separate port from SASP)
- Obtains server topology information and workload balancing recommendations from each target system and for each target application

➤ **Configuration**

- Must identify all eligible load balancing agents (by source IP address and source port)
- Must identify all eligible load balancers by source IP address
- IP address and port it should listen to (application specific dynamic DVIPA strongly encouraged)
- Other parameters (debug levels, polling interval, WLM recommendations, etc.)

z/OS Load Balancing Advisor (*continued*)

➤ System vs. server-specific WLM recommendation types

- ┆ Configurable as a global default (wlm statement)
 - Values for wlm statement are basewlm or serverwlm
 - Default value is basewlm, meaning, use system WLM recommendations
 - Chosen as the default for compatibility with pre-V1R7 systems
 - serverwlm value means use server-specific WLM recommendations
 - May be overridden on a port number basis (port_list statement)
- ┆ System WLM recommendations
 - Reflect the displaceable capacity of the z/OS system relative to other z/OS systems in the Sysplex
- ┆ Server-specific WLM recommendations
 - Reflect how well the application is meeting its WLM goals
 - How much displaceable capacity is available on the target system at the importance level of the application
 - Only available with z/OS V1R7
- ┆ Choosing WLM recommendation types
 - Server-specific (serverwlm) recommendations are recommended for most applications
 - Provide more granular, more accurate load balancing information
 - Some applications better suited for system WLM recommendations (basewlm)
 - Those that serve as an access point to applications, which run in a separate address space (and therefore, a different service class) - such as TN3270, INETD, and FTPD
 - If FTP servers run in the same service class as the FTP daemon, use server-specific WLM recommendations (serverwlm) - however, if FTP servers run in a different service class from the FTP daemon, then use system WLM recommendations (basewlm)

Advisor configuration information

Sample is in `hlq.SEZAINST(EZBLBADC)` (alias `LBADVCNF`)

> **agent_connection_port**

Specifies the port the advisor should listen on for connections from agents

> **agent_id_list**

Specifies which agents are allowed to connect to the advisor

> **debug_level** (optional)

Specifies the level of debug information that will be logged. Default 7 (Error, Warning, Event).

> **lb_connection_v4 and/or lb_connection_v6**

Specifies the local IPv4 or IPv6 address and port the advisor should listen on for connections from load balancers. Use both to listen for either type of connection. Default port value is 3860.

> **lb_id_list**

Specifies remote addresses from which load balancers are allowed to connect to the advisor

> **port_list** (optional)

Specifies a list of ports and the type of WLM server recommendation that should be used for each. Overrides value from the `wlm` statement on a port basis. Only valid for V1R7.

> **update_interval** (optional)

Specifies how often agents update the advisor with new information. May also determine how often the advisor updates the load balancer if the load balancer supports the 'push' flag. Default 60 seconds.

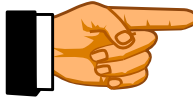
Advisor configuration information (*continued*)

➤ **wlm** (optional)

- Specifies the default type of WLM recommendations that will be attempted. Choices are **serverwlm** and **basewlm**. Default is **basewlm**. Only valid for V1R7.

➤ **lb_id_list**

- If the load balancer has multiple source IP addresses it can use, make sure **lb_id_list** contains the address the load balancer will use as a source IP address when connecting as a SASP client. If your load balancer-to-advisor connection is failing, examine the advisor log for a message with the text, 'Unauthorized connection attempt from <ip_address>'. If this message is present, <ip_address> is the address the load balancer is using as a source IP address for connecting to the advisor. Insert this address into the **lb_id_list** statement, restart the advisor and reconnect from the load balancer.
- For CISCO CSM, this is the client VLAN interface IP address, not the server VLAN IP address.



➤ **General configuration hints:**

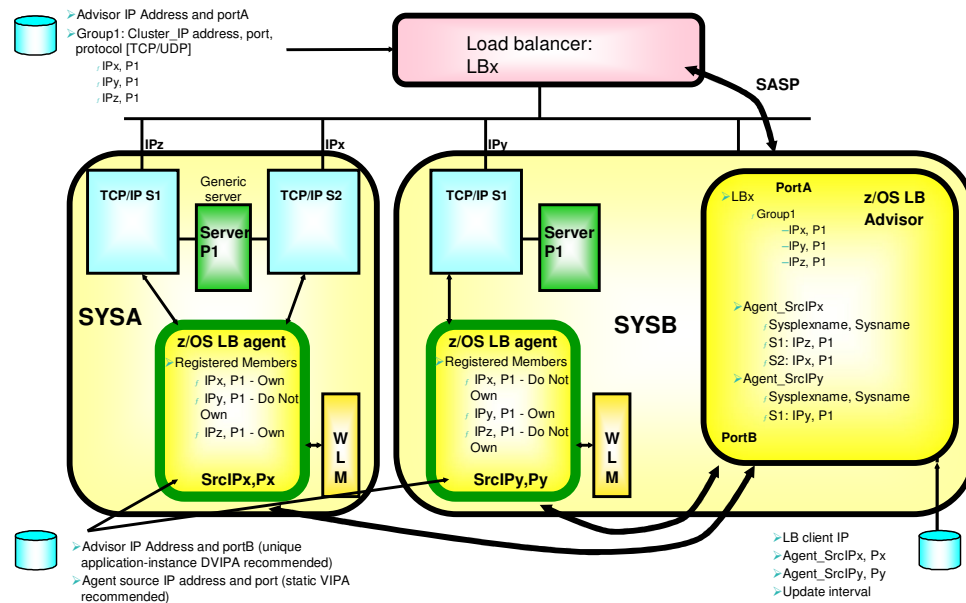
- Any statement that contains the phrase "id" specifies a remote endpoint.
- Any statement that contains the phrase "connection" specifies a local IP address and possibly port.
- These are true for both the advisor and agent configuration files.

Advisor configuration example

```
debug_level      15          # Error, Warning, Event, Info
update_interval  120        # Agent updates every 2 minutes
lb_connection_v4 10.67.5.1..3860 # DVIPA advisor listen endpoint
lb_id_list       {          # LBs we will talk with
  10.67.1.11     # SDBAV4
}
agent_connection_port 8100 # Agents connect to Advisor on this port
agent_id_list     {          # Agents we will talk with
  10.67.1.1..8000 # SD1AV4
  10.67.1.2..8000 # SD2AV4
  10.67.30.22..8000 # SD2A2V4
  10.67.1.10..8000 # SDAAV4
}

wlm serverwlm    # Request server-specific WLM weights
port_list        {
  21 wlm basewlm # Use system WLM weights for FTP
}
```

z/OS LB agent address spaces - configuration overview



z/OS load balancing agent

> z/OS LB agent

- New, stand-alone application
 - Started via an MVS started task
 - Accepts MVS operator commands for display and modification purposes
- Executes on every target system in the Sysplex
 - Or at least on every system in the Sysplex that is a target of a load balanced request
 - Provides load balancing advice for specified TCP/UDP server applications on local system
 - Only one active instance allowed per MVS system
- Supports multiple TCP/IP stacks and all known server types: stack-affinity, generic, bind-specific, shareport groups etc.
 - Computes weights based on WLM, server availability, server health (dropped connections due to backlog queue full or dropped datagrams due to UDP queue limit exceeded)
 - Communications Server weight (CS weight) represents server health - similar to Sysplex Distributor's Server accept Efficiency Fraction (SEF)
 - Server-specific WLM weights (V1R7) or system WLM weights
- Simple configuration
 - Specify IP address and port for load balancing advisor
 - Specify source IP address/port to be used in connecting to advisor
 - Static VIPA recommended for single stack systems (allows for failures in physical interfaces)
 - For CINET, unique application-instance DVIPA recommended
 - The same source port can be used by all agents (simplifies configuration)
 - Optionally specify the debug level

Agent configuration information

Sample is in hlq.SEZAINST(EZBLBAGC) (alias LBAGECNF)

➤ **advisor_id**

Specifies the remote IP address and port of the Advisor this agent will communicate with

➤ **host_connection**

Specifies the local IP address and port the agent will bind to for communications with the advisor.

➤ **debug_level** (optional)

Specifies the level of debug information that will be logged. Default 7 (Error, Warning, Event).

Agent configuration examples

Agent #1:

```
debug_level      15                # Error, Warning, Event, Info
advisor_id       10.67.5.1..8100    # Advisor IP and port
host_connection  10.67.1.2..8000    # Agent source IP and port
```

=====

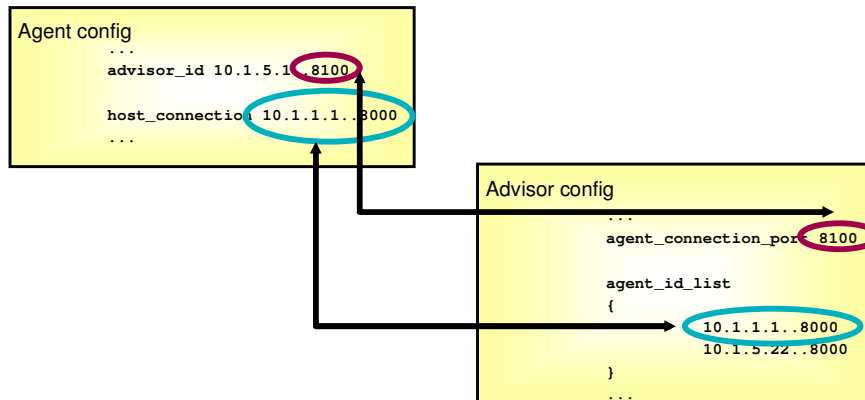
Agent #2:

```
debug_level      15                # Error, Warning, Event, Info
advisor_id       10.67.5.1..8100    # Advisor IP and port
host_connection  10.67.30.22..8000 # Agent source IP and port
```

Configuration hints and tips

➤ Configuration relationships among the advisor and agent

- See connected arrows
- IP address in `advisor_id` statement can be any IP address belonging to the TCP/IP stack the advisor is running on, however, it is recommended this be a unique application-instance DVIPA.



Notes - Advisor Started Procedure Sample (in SEZAINST, alias EZBLBADV)

NOTES

```
//LBADV PROC
/* IBM Communications Server for z/OS
/* SMP/E distribution name: EZBLBADV
/* Licensed Materials - Property of IBM
/* (C) copyright IBM Corp. 2004
/* Function: Sample procedure for running the
/* z/OS Load Balancing Advisor
/*
//LBADV EXEC PGM=EZBLBADV,REGION=0K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON)'/
/*** Notes:
/* - The system link list concatenation must contain the TCP/IP
/* runtime libraries and the C runtime libraries. If they are
/* not in the link list concatenation, this procedure will need
/* to be changed to STEPLIB to them.
/* If you add them to STEPLIB, they must be APF authorized.
/* - The z/OS Load Balancing Advisor requires a configuration file
/* which can be a member of an MVS PDS(E), an MVS sequential file,
/* or an HFS file.
/*
//CONFIG DD DSN=TCPIP.TCPPARMS(LBADVCNF),DISP=SHR
/*CONFIG DD DSN=TCPIP.CONFIG.LBADV,DISP=SHR
/*CONFIG DD PATH='/etc/lbadv.conf',PATHOPTS=(ORDONLY)
//STDENV DD DUMMY
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD DUMMY
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//CEESNAP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
> //SYSDUMP DD DISP=SHR,DSN=your.data.set.name
```

Notes - Agent Started Procedure Sample (in SEZAINST, alias EZBLBAGE)

NOTES

```
//LBAGENT PROC
/* IBM Communications Server for z/OS
/* SMP/E distribution name: EZBLBAGE
/* Licensed Materials - Property of IBM
/* (C) Copyright IBM Corp. 2004
/* Function: Sample procedure for running the
/* z/OS Load Balancing Agent
/*
//LBAGENT EXEC PGM=EZBLBAGE,REGION=0K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON)'/
/*** Notes:
/* - The system link list concatenation must contain the TCP/IP
/* runtime libraries and the C runtime libraries. If they are
/* not in the link list concatenation, this procedure will need
/* to be changed to STEPLIB to them.
/* If you add them to STEPLIB, they must be APF authorized.
/* - The z/OS Load Balancing Agent requires a configuration file
/* which can be a member of an MVS PDS(E), an MVS sequential file,
/* or an HFS file.
/*
//CONFIG DD DSN=TCPIP.TCPPARMS(LBAGECNF),DISP=SHR
/*CONFIG DD DSN=TCPIP.CONFIG.LBAGENT,DISP=SHR
/*CONFIG DD PATH='/etc/lbagent.conf',PATHOPTS=(ORDONLY)
//STDENV DD DUMMY
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN DD DUMMY
//SYSERR DD SYSOUT=*
//SYSOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//CEESNAP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
//SYSDUMP DD DISP=SHR,DSN=your.data.set.name
```

Logging

- **Advisor and Agent both log to syslogd, daemon facility**
- **Syslogd must be configured and started before starting the Advisor or Agent**
- **Default debug level (7) should normally be used unless problem documentation needs to be gathered**
 - ┆ Increased levels of debug may result in voluminous amounts of information
- **See the following "Notes" page for a description of the debug levels and log snippets. Agent log entries are similar to Advisor log entries.**

Notes - Debug levels

NOTES

➤ Logging levels for Advisor and Agent

- 0 None. No debug messages are logged.
- 1 Error-level messages are logged.
- 2 Warning-level messages are logged.
- 4 Event-level messages are logged.
- 8 Info-level messages are logged.
- 16 Message level messages are logged. These are details of the messages (packets) sent between the Advisor and LB, and the Advisor and Agent.
- 32 Collection-level messages are logged. These are details of the collection and manipulation of data supporting the calculated weights. This level only has meaning to the Agent. The Advisor does not log any data at this level.
- 64 Debug level messages are logged. These are internal debug messages intended for Development and Service.
- 128 Trace level messages are logged. These are function entry and exit traces that show the path through the code.

➤ Guideline: To log a combination of debug levels, add the debug level numbers. The default debug level is 7, which captures all Error, Warning, and Event messages.

➤ Anatomy of log a message (following the normal prefix that syslogd adds):

```
<debug_level>:<thread_id>:<procedure_name>:<message>
```

Notes - Samples of Advisor logging

NOTES

- > For the Advisor, each line has the following type of prefix:
 - !Apr 6 13:27:57 MVSWS lbadv[67108888]: <message text>
- > Sample lines follow using debug value of 127 showing <message text>, without the prefix:
 - !Debug 127= Error, Warning, Event, Info, Message, Collection, Debug
 - !Format is <debug_level>:<thread_id>:<procedure_name>:<message>
- > EVENT :00:main:EZD1231I LBADV STARTING
- > INFO :00:main:Compiled on Nov 19 2004 at 16:41:15
- > WARNING:00:lbap_arm:LBADV ARM register request unsuccessful, return code C, reason code 168
- > INFO :00:main:Configuration file: /etc/lbadv.conf
- > INFO :00:main:Source of data: start option
- > INFO :00:lb_read_file_contents:Start processing config file '/etc/lbadv.conf'
- > INFO :00:lb_process_config_stmt:Processing statement: 'DEBUG_LEVEL 127'
- > INFO :00:lb_process_config_stmt:Processing statement: 'update_interval 30'
- > INFO :00:lb_process_config_stmt:Processing statement: 'lb_connection_v4 0.0.0.0..3860'
- > INFO :00:lb_process_config_stmt:Processing statement: 'agent_connection_port 3333'
- > INFO :00:lb_process_config_stmt:Processing statement: 'lb_id_list'
- > INFO :00:lb_process_config_stmt:Processing statement: 'agent_id_list'
- > INFO :00:lb_process_config_stmt:Processing statement: 'wlm basewlm'
- > INFO :00:lb_process_config_stmt:Processing statement: 'port_list'
- > INFO :00:lb_process_config_stmt:Processing statement: ' wlm basewlm'
- > INFO :00:lb_process_config_stmt:Processing statement: ' wlm basewlm'
- > INFO :00:lb_process_config_stmt:Processing statement: ' wlm serverwlm'
- > INFO :00:lb_process_config_stmt:Processing statement: ' wlm basewlm'
- > INFO :00:lb_read_file_contents:Finished processing config file '/etc/lbadv.conf'

Notes - Samples of Advisor logging (Cont.)

NOTES

```
>DEBUG :00:main:Select returned
>DEBUG :00:main:Received I/O on a listening socket
>DEBUG :00:main:Connection from an Agent
>INFO :00:main:Agent connection from :ffff:127.0.0.1, port 6666
>DEBUG :00:main:Creating Agent connection thread
>INFO :00:lbap_thread_create:Agent Connection thread created for tid 1675513000000002
>DEBUG :00:lm_add_conn_thread_id:Adding thread, 1675513000000002
>DEBUG :00:lm_add_conn_thread_id:WRITE LOCKING 1641F1C4 ./lmmain.c 2494
>EVENT :02:lm_service_agent_connection:Agent Connection thread active
>DEBUG :02:lm_service_agent_connection:WRITE LOCKING 1641F1AC ./lmagnt.c 260
>DEBUG :00:lm_add_conn_thread_id:WRITE LOCKED 1641F1C4 ./lmmain.c 2494
>DEBUG :02:lm_service_agent_connection:WRITE LOCKED 1641F1AC ./lmagnt.c 260
>DEBUG :02:lm_add_agent:TESTING WRITE LOCK 1641F1AC ./lmagnt.c 650
>DEBUG :00:lm_add_conn_thread_id:WRITE UNLOCKED 1641F1C4 ./lmmain.c 2504
>DEBUG :00:main:Remainder of select() timeout is 0 seconds
>DEBUG :00:main:Select waiting for data to be received
>DEBUG :02:lm_add_agent:WRITE WAS ALREADY LOCKED 1641F1AC ./lmagnt.c 650
>DEBUG :02:lm_service_agent_connection:WRITE UNLOCKED 1641F1AC ./lmagnt.c 262
>DEBUG :02:lm_service_agent_connection:READ LOCKING 1641F1AC ./lmagnt.c 271
>DEBUG :02:lm_service_agent_connection:READ LOCKED 1641F1AC ./lmagnt.c 271
>DEBUG :02:lm_service_agent_connection:WRITE LOCKING 1641FD18 ./lmagnt.c 272
>DEBUG :02:lm_service_agent_connection:WRITE LOCKED 1641FD18 ./lmagnt.c 272
>DEBUG :02:clear_immed_action_msgs:TESTING WRITE LOCK 1641F1AC ./lmactmsg.c 308
>DEBUG :02:clear_immed_action_msgs:WRITE WAS ALREADY LOCKED 1641F1AC ./lmactmsg.c 308
>DEBUG :02:clear_immed_action_msgs:TESTING WRITE LOCK 1641FD18 ./lmactmsg.c 309
>DEBUG :02:clear_immed_action_msgs:WRITE WAS ALREADY LOCKED 1641FD18 ./lmactmsg.c 309
>DEBUG :02:clear_immed_action_msgs:WRITE LOCKING 1641F1F4 ./lmactmsg.c 336
>DEBUG :02:clear_immed_action_msgs:WRITE LOCKED 1641F1F4 ./lmactmsg.c 336
>DEBUG :02:lm_find_immed_action_msg_id:TESTING WRITE LOCK 1641F1F4 ./lmactmsg.c 240
>DEBUG :02:lm_find_immed_action_msg_id:WRITE WAS ALREADY LOCKED 1641F1F4 ./lmactmsg.c 240
>DEBUG :02:lm_find_immed_action_msg_id:Did not find immediate action message id
>DEBUG :02:clear_immed_action_msgs:No MVS console immediate action message found to clear
>DEBUG :02:clear_immed_action_msgs:WRITE UNLOCKED 1641F1F4 ./lmactmsg.c 364
```

Notes - Samples of Advisor logging (Cont.)

NOTES

```

>MESSAGE:02:lm_zap_receive_decode_validate:-----
>MESSAGE:02:lm_zap_receive_decode_validate:ZAPHeaderTLV
>MESSAGE:02:lm_zap_receive_decode_validate:-----
>MESSAGE:02:lm_zap_receive_decode_validate:type = 0x 2F10
>MESSAGE:02:lm_zap_receive_decode_validate:len = 0x 0009
>MESSAGE:02:lm_zap_receive_decode_validate:pv = 0x 02
>MESSAGE:02:lm_zap_receive_decode_validate:msglen = 0x 0000001F
>MESSAGE:02:lm_zap_receive_decode_validate:-----
>MESSAGE:02:lm_zap_receive_decode_validate:AgentRegistrationRequestTLV
>MESSAGE:02:lm_zap_receive_decode_validate:-----
>MESSAGE:02:lm_zap_receive_decode_validate:type = 0x 1F10
>MESSAGE:02:lm_zap_receive_decode_validate:len = 0x 0016
>MESSAGE:02:lm_zap_receive_decode_validate:systemNameLen = 0x 08
>MESSAGE:02:lm_zap_receive_decode_validate:systemName = 0x
>MESSAGE:02:lm_zap_receive_decode_validate:D4E5E2E6 00404040
>MESSAGE:02:lm_zap_receive_decode_validate:sysplexNameLen = 0x 08
>MESSAGE:02:lm_zap_receive_decode_validate:sysplexName = 0x
>MESSAGE:02:lm_zap_receive_decode_validate:D3D6C3C1 D3404040
>DEBUG :02:lm_process_agentRegistrationRequest:TESTING WRITE LOCK 1641F1AC ./lmagnt.c 1271
>DEBUG :02:lm_process_agentRegistrationRequest:WRITE WAS ALREADY LOCKED 1641F1AC ./lmagnt.c 1271
>DEBUG :02:lm_process_agentRegistrationRequest:TESTING WRITE LOCK 1641FD18 ./lmagnt.c 1272
>DEBUG :02:lm_process_agentRegistrationRequest:WRITE WAS ALREADY LOCKED 1641FD18 ./lmagnt.c 1272
>DEBUG :02:lm_process_agentRegistrationRequest:ZAP Message, AgentRegistrationRequest, OK while in Initial state
>EVENT :02:lm_process_agentRegistrationRequest:EZD12611LBADV AGENT CONNECTED FROM ::ffff:127.0.0.1
>DEBUG :02:lm_send_zap_message:TESTING WRITE LOCK 1641F1AC ./lmagnt.c 3154
>DEBUG :02:lm_send_zap_message:WRITE WAS ALREADY LOCKED 1641F1AC ./lmagnt.c 3154
>DEBUG :02:lm_send_zap_message:TESTING WRITE LOCK 1641FD18 ./lmagnt.c 3155
>DEBUG :02:lm_send_zap_message:WRITE WAS ALREADY LOCKED 1641FD18 ./lmagnt.c 3155

```

TCP/IP Profile example

```
IPCONFIG
  DATAGRAMFWD                ; Enable datagram forwarding
  DYNAMICXCF 10.67.30.1 255.255.255.0 1 ; Enable dynamic XCF
VIPADYNAMIC
  VIPARANGE DEFINE 255.255.255.0 10.67.5.1
ENDVIPADYNAMIC

DEVICE VIPA41  VIRTUAL 0      ; Static VIPA
LINK  LVIPA41  VIRTUAL 0  VIPA41
HOME  10.67.1.1                LVIPA41

PORT
  3860 TCP LBADV                ; SASP Workload Advisor (LB connections)
  8100 TCP LBADV                ; SASP Workload Advisor (Agent connections)
  8000 TCP LBAGENT              ; SASP Workload Agent (Advisor connection)
```

- **These statements can be duplicated in all TCP/IP stack profiles in the Sysplex**
- ✓ Simplifies port reservation for this function
 - ✓ Allows advisor to be restarted on any system in the Sysplex

Security and control considerations

➤ Define OMVS segments if not already defined

➤ Configure RACF

- / Define user IDs to RACF, associate with OMVS segment, and assign UIDs
- / Add default users to the STARTED class
- / Optionally, restrict which users can start the applications
 - Only one instance of the advisor is permitted per Sysplex
 - Only one instance of the agent is permitted per MVS system
- / Permit access to the BPX.WLMSEVER resource profile
 - For RACF, explicit access not required if the resource profile is not defined
 - For other security products, consult the product documentation as to whether explicit access is required for your installation
- / Sample RACF definitions can be found in hlq.SEZAINST(EZARACF)
 - Look for "LBADV" and "LBAGENT"

RACF configuration notes

NOTES

- > Sample RACF definitions can be found in hlq.SEZAINST(EZARACF)
 - ┆ Look for "LBADV" and "LBAGENT"
- > Both require OMVS segments (UID may be zero or non-zero)
 - ADDUSER LBADV DFLTGRP (OMVSGRP) OMVS (UID (nn) HOME ('/'))
 - ADDUSER LBAGENT DFLTGRP (OMVSGRP) OMVS (UID (nn) HOME ('/'))
- > Add the default users to the started class
 - RDEFINE STARTED LBADV.* STDATA (USER (LBADV))
 - RDEFINE STARTED LBAGENT.* STDATA (USER (LBAGENT))
- > If you would like to control which users can start the applications:
 - ┆ Ensure that the OPERCMDS class is active and RACLISTed and RACLIST processing is enabled.
 - SETROPTS CLASSACT (OPERCMDs)
 - SETROPTS RACLIST (OPERCMDs)
 - ┆ Define the following OPERCMDS class profiles using a security product like RACF
 - RDEFINE OPERCMDS (MVS.SERV MGR.LBADV) UACC (NONE)
 - RDEFINE OPERCMDS (MVS.SERV MGR.LBAGENT) UACC (NONE)
 - ┆ Grant the Advisor access to the OPERCMDS class.
 - PERMIT MVS.SERV MGR.LBADV CLASS (OPERCMDs) ACCESS (CONTROL) -
ID (userid)
 - PERMIT MVS.SERV MGR.LBAGENT CLASS (OPERCMDs) ACCESS (CONTROL) - ID (userid)
 - ┆ Refresh the OPERCMDS class.
 - SETROPTS RACLIST (OPERCMDs) REFRESH.

Security and control considerations

➤ Make Advisor and Agent non-swappable

• V1R7

– Both will run non-swappable by default. No action required.

• V1R4, V1R5, and V1R6

– Configuration of the Program Properties Table (PPT) in the SCHEDxx member is required in order to run non-swappable.

```

PPT PGMNAME(EZBLBAGE)      /* z/OS LOAD BALANCING AGENT  */
KEY(8)                    /* PROTECTION KEY              */
NOSWAP                    /* NONSWAPPABLE                */
PRIV                      /* PRIVILEGED                   */
SYST                      /* SYSTEM TASK , NOT TIMED     */
AFF(NONE)                 /* NO CPU AFFINITY             */
PPT PGMNAME(EZBLBADV)     /* Z/OS LOAD BALANCING ADVISOR */
KEY(8)                    /* PROTECTION KEY              */
NOSWAP                    /* NONSWAPPABLE                */
PRIV                      /* PRIVILEGED                   */
SYST                      /* SYSTEM TASK , NOT TIMED     */
• AFF(NONE)               /* NO CPU AFFINITY             */

```

➤ Ensure the Advisor and Agents receive the proper dispatching priority

• Verify Advisor and Agents are assigned to the WLM SYSSTC service class

• See "MVS Planning: Workload Management" for more information



Recovery scenarios

Automatic restart of advisor/agent

➤ **Ensure that automation is in place to restart**

- ┆ The advisor
 - An the same/other system in the Sysplex in cases of failures (of the advisor or the system)
- ┆ The agent
 - In the same system when the agent terminates abnormally

➤ **Can be accomplished with ARM (Automatic Restart Manager) policy or other automation**

- ┆ Note that while AUTOLOG can be used to start the agent, it can NOT be used to monitor the availability of the agent after initial startup.

➤ **Only one advisor can be active per Sysplex and only one agent per system**

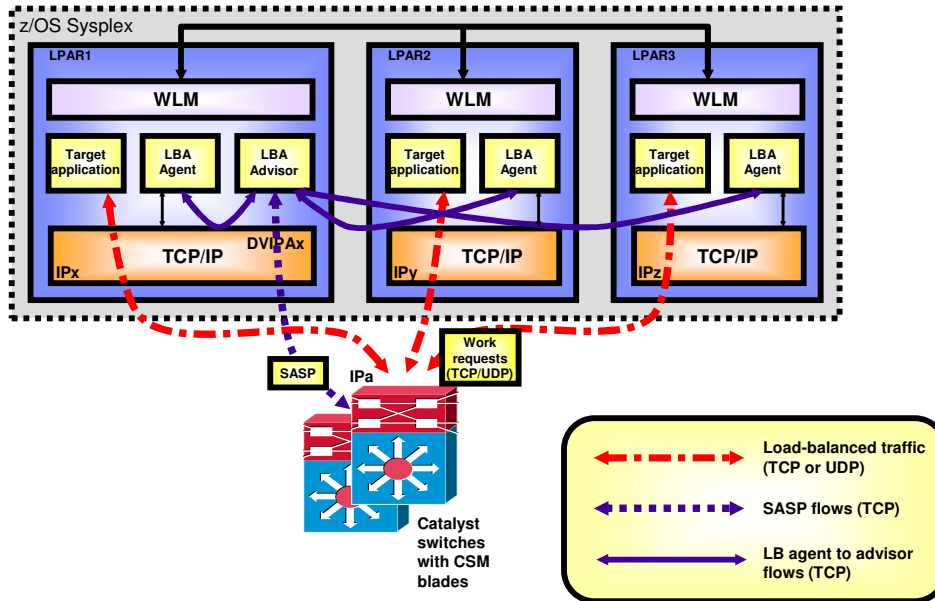
- ┆ Note: Internal checks will prevent the starting of multiple advisors (within the Sysplex) or multiple agents within the same system.

➤ **Special recovery considerations for the advisor on a multi-stack system (CINET)**

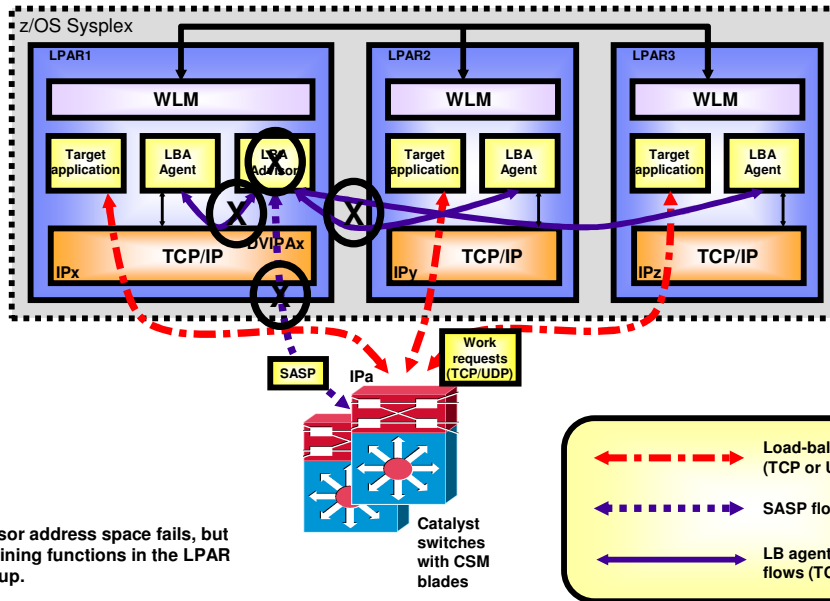
- ┆ Refer to "CINET Considerations" later in this presentation

➤ **Advisor may be (re)started while agents already running or vice versa**

Sample setup - normal operations

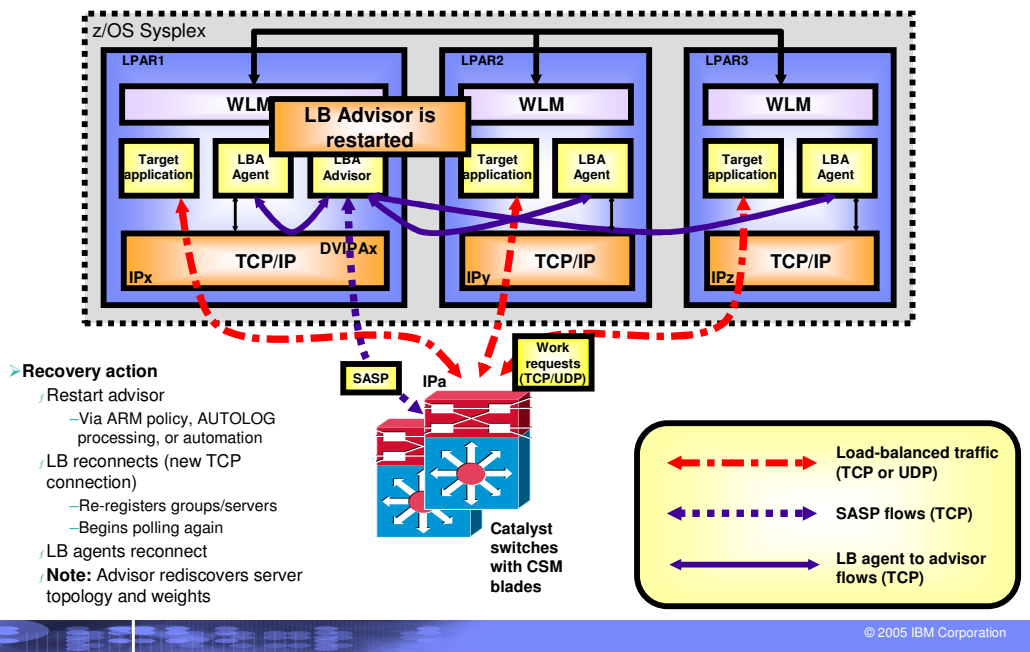


Sample setup - recovery scenario A - advisor fails

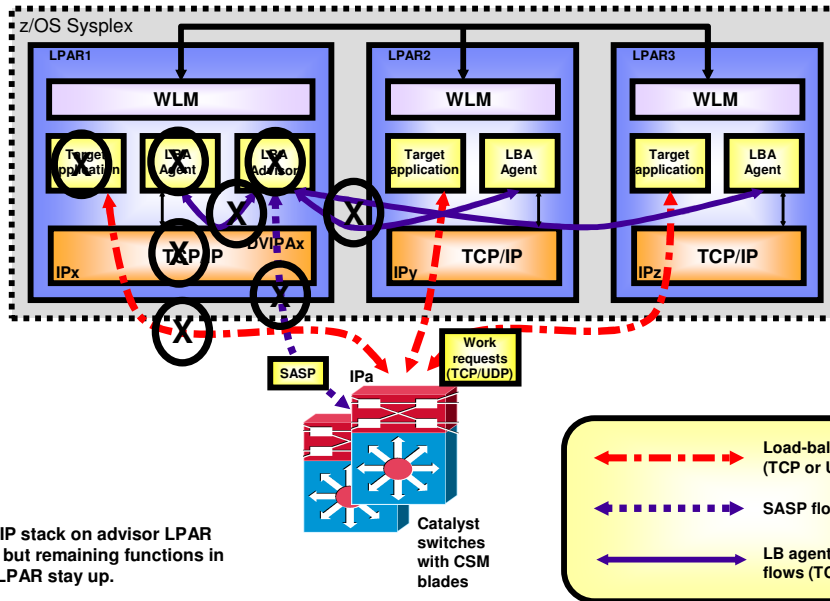


> Advisor address space fails, but remaining functions in the LPAR stay up.

Sample setup - recovery scenario A - advisor fails (continued)

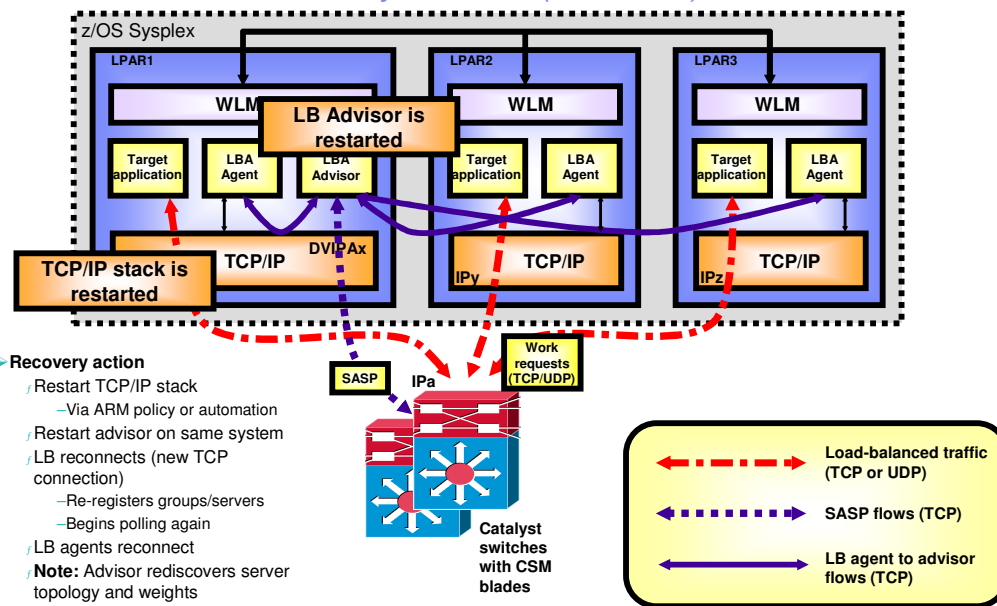


Sample setup - recovery scenario B TCP/IP stack on advisor system fails

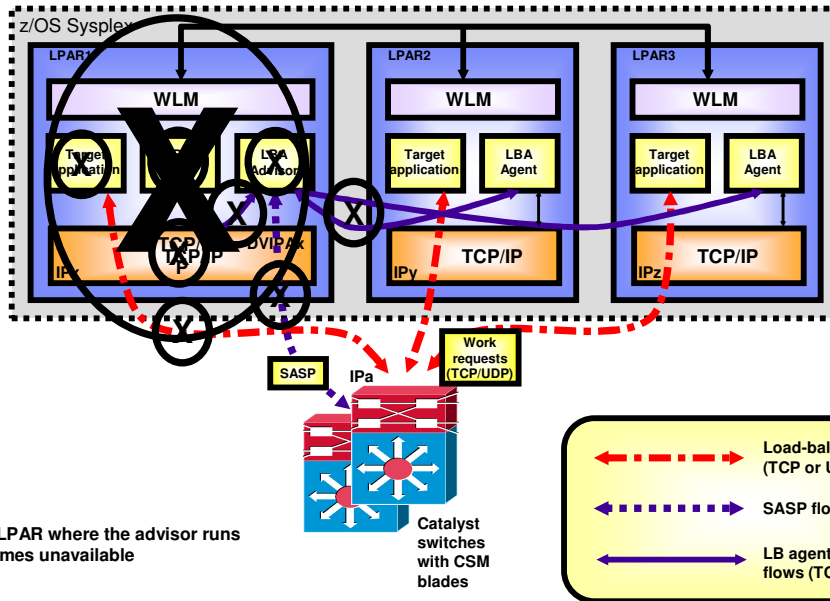


> TCP/IP stack on advisor LPAR fails, but remaining functions in that LPAR stay up.

Sample setup - recovery scenario B TCP/IP stack on advisor system fails (continued)

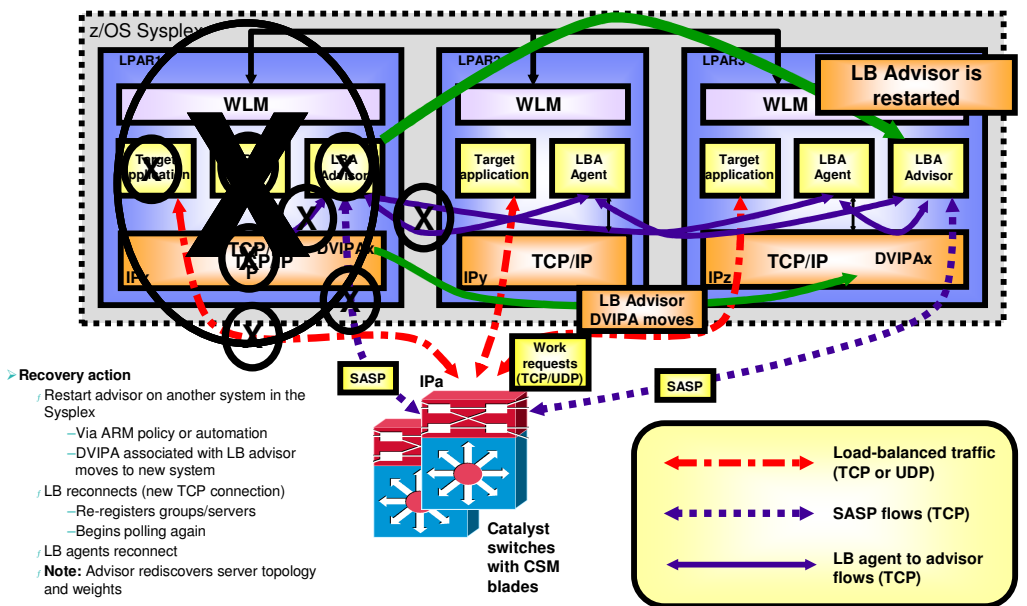


Sample setup - recovery scenario C - advisor system fails

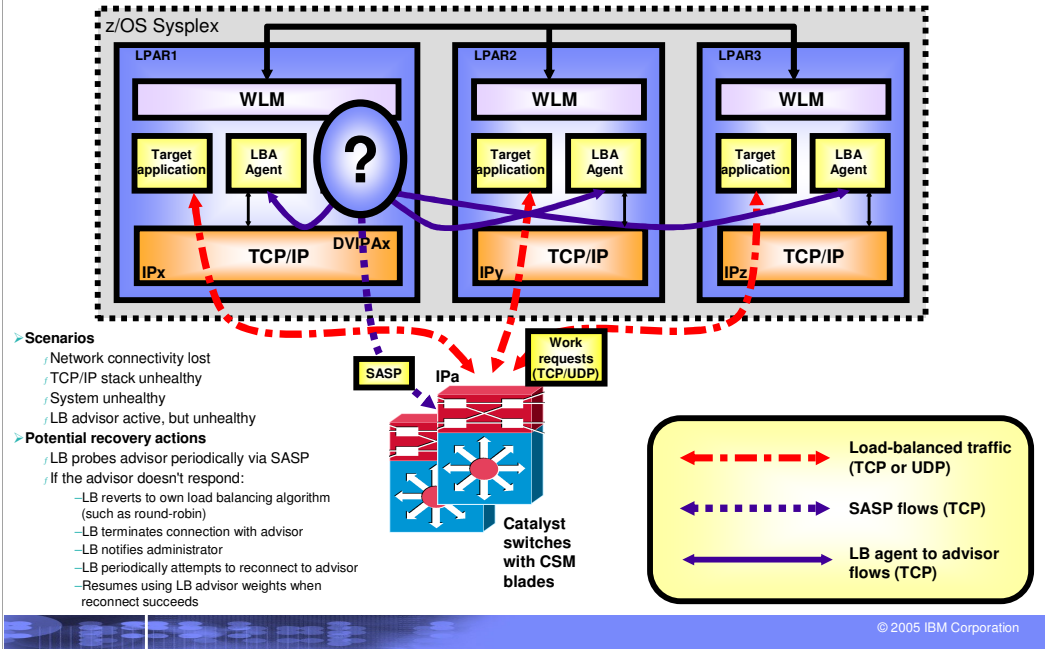


> The LPAR where the advisor runs becomes unavailable

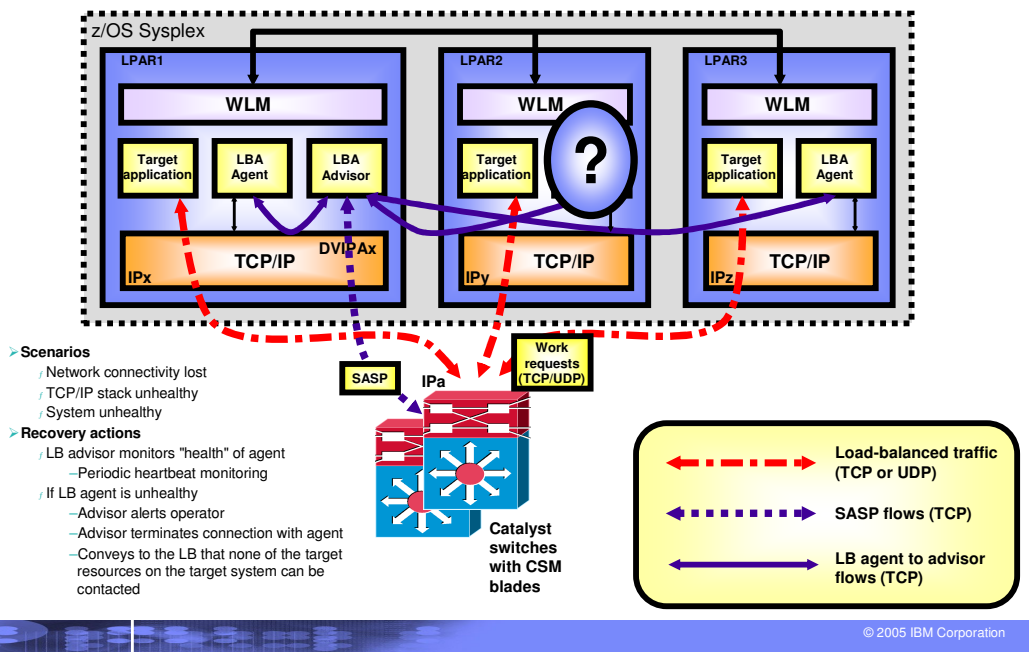
Sample setup - recovery scenario C - advisor system fails (continued)



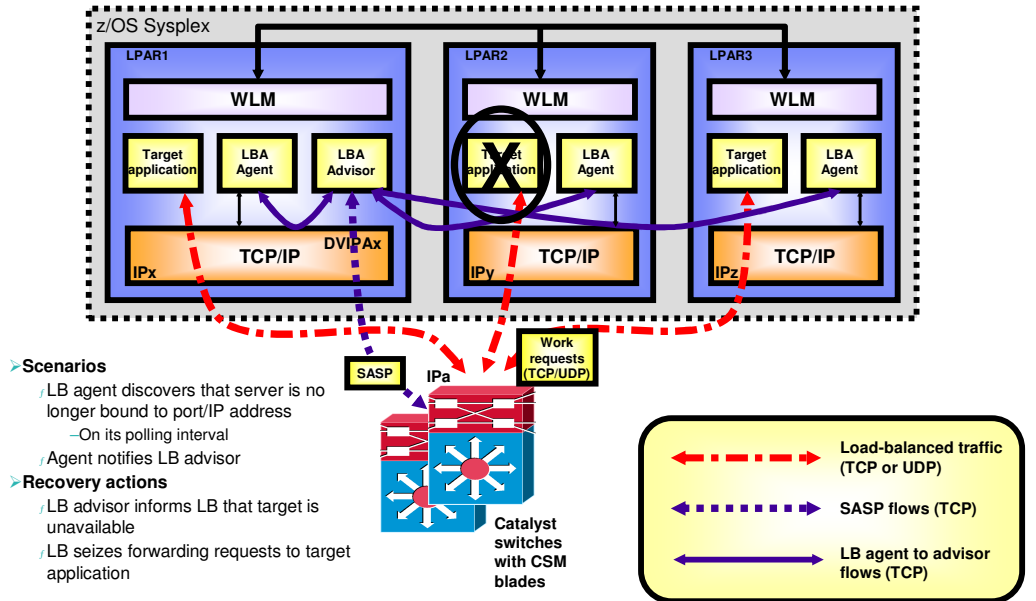
Sample setup - recovery scenario D - advisor not responding



Sample setup - recovery scenario E - agent not responding



Sample setup - recovery scenario F - target application fails



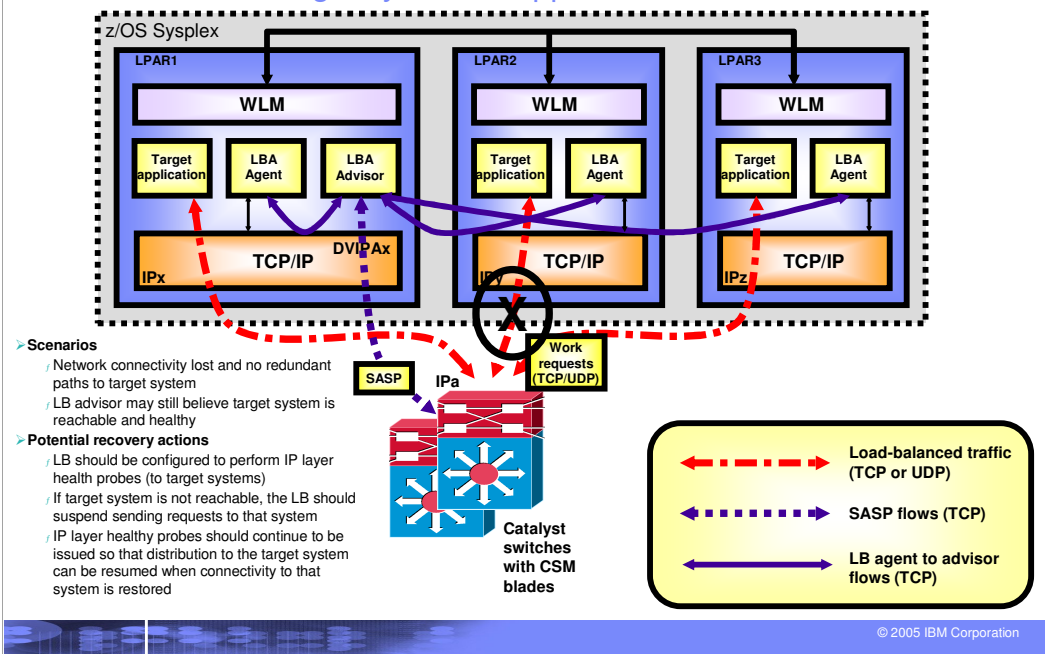
>Scenarios

- LB agent discovers that server is no longer bound to port/IP address
 - On its polling interval
- Agent notifies LB advisor

>Recovery actions

- LB advisor informs LB that target is unavailable
- LB seizes forwarding requests to target application

Sample setup - recovery scenario G - loss of network connectivity between LB and target system or application





Trademarks, Copyrights and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Trvoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e/logo/business	DB2	iSeries	OS/400	xSeries
ADX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR

NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2005. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.