

z/OS V1R13

IBM Health Checker for z/OS: New and updated health checks

Session objectives

- Very briefly describe what Health Checker, a health check, and a migration check is
- Present an overview of new and changed health checks developed in z/OS® V1R13 by
 - Allocation
 - Comm Server
 - Consoles
 - Device Manager
 - Supervisor
 - zFS
 - VSAM RLS
 - USS
 - XCF

Overview

- **Problem Statement / Need Addressed:**
 - Continuous need for programmatic approach to ensuring component best practices and to providing migration assistance
- **Solution:**
 - Provide additional health checks for existing and new best practices, and for migration actions
- **Benefit:**
 - Improved system health and smoother release migration

Overview – What is Health Checker?

- IBM Health Checker for z/OS is a component of MVS that **identifies potential problems before they impact your installation's availability** or, in worst cases, would cause outages.
- It checks the current active z/OS and Sysplex settings and definitions for a system and **compares the values to those suggested** by IBM or defined by you.
- It produces output in the form of **detailed messages** to let you know of both **potential problems** and **suggested actions** to take.

Overview – What are health checks?

- A health check is doing the actual “inspection of settings...”, as a “plug-in” to the Health Checker framework
- There are many health checks
 - from individual IBM components or ISVs
 - each check is supposed to focus on one scenario only
- Health checks are named (*CheckOwner, CheckName*)
 - *CheckOwner* is typically *CompanyName_ComponentName*
 - *CheckName* is typically *ComponentName_CheckPurpose*
- For example: (IBMASM, ASM_PLPA_COMMON_USAGE)
 - An IBM Auxiliary Storage Manager (ASM) check, which "Looks at the slot usage of the PLPA and Common page data sets"

Overview – What are migration checks?

- Migration checks are health checks which automate the task of identifying the need for migration actions
- Migration checks are shipped inactive and are run on demand before an actual migration
- A common naming convention for migration checks is (*CheckOwner, ZOSMIGxxxRestOfCheckName*), where xxx can be a specific release or just REC (recommended migration action without a specific release, yet...)
- For example:
 - (IBMCNZ, ZOSMIGV1R13_CNZ_Cons_Oper_Mode)
 - (IBMZFS, ZOSMIGREC_ZFS_RM_MULTIFS)

Overview – List of new (or updated*) health checks

Allocation (LI2895)	(IBMALLOC, ALLOC_ALLC_OFFLN_POLICY) (IBMALLOC, ALLOC_SPEC_WAIT_POLICY) (IBMALLOC, ALLOC_TIOT_SIZE)
Comm Server	(IBMCS, CSRES_AUTOQ_GLOBALTCPDATA) (IBMCS, CSRES_AUTOQ_RESOLVEVIA) (IBMCS, CSRES_AUTOQ_TIMEOUT) (IBMCS, CSVTAM_VIT_SIZE) * n/a for R13+ anymore (IBMCS, CSVTAM_DSPSIZE) * n/a for R13+ anymore
Consoles (LI2073)	(IBMCNZ, CNZ_Console_Operating_Mode) (IBMCNZ, ZOSMIGV1R13_CNZ_Cons_Oper_Mode)
Device Manager	(IBMDMO, DMO_TAPE_LIBRARY_INIT_ERRORS)

(LI2414)	
Supervisor	(IBMSUP, SUP_HiperDispatchCPUConfig) *
zSeries® File System (LI2615)	(IBMZFS, ZOSMIGV1R13_ZFS_FILESYS)
VSAM RLS (LI28483)	(IBMVSAMRLS, VSAMRLS_QUIESCE_STATUS)
z/OS Unix	(IBMUSS, USS_PARMLIB) *
XCF (OA34339)	(IBMXCF, XCF_SFM_CFSTRHANGTIME) *

Interactions and dependencies

- Software Dependencies
 - Have Health Checker started
- Hardware Dependencies
 - None
 - Any new checks which might target new hardware are presented separately
- Exploiters
 - Operators and System Programmers

Migration and coexistence considerations

- Look for pre-R13 APARs for the following checks
 - (IBMCNZ, ZOSMIGV1R13_CNZ_Cons_Oper_Mode) OA32930
 - (IBMZFS, ZOSMIGV1R13_ZFS_FILESYS) OA35465
 - (IBMSUP, SUP_HiperDispatchCPUConfig) OA35059
 - (IBMXCF, XCF_SFM_CFSTRHANGTIME) OA34439
 - (IBMVSAMRLS, VSAMRLS_QUIESCE_STATUS) OA28483
- For changed checks, if shipped with a new DATE, you will have to re-evaluate any POLICY statements that apply, to make sure that those policies continue to get applied and not being rejected as “outdated”
 - Updated health check (IBMXCF, XCF_SFM_CFSTRHANGTIME) has a new DATE

Installation

- These IBM health checks are automatically installed with their associated product or component, or are shipped with the base operation system for the new release
- In addition some checks, in particular migration checks, will have APARs available for previous releases

Check details – Allocation

- (IBMALLOC, ALLOC_ALLC_OFFLN_POLICY)
- Checks the value of the ALLC_OFFLN policy (to be followed when an allocation request needs a device that is offline, or must wait for a non-specific volume or unit) in the current Allocation settings.
- This check ensure that the best ALLC_OFFLN policy allocation setting for the particular environment is set. Certain ALLC_OFFLN policy values can result in a deadlock.
- Check parameter 'POLICY(policyvalue_1[,...,policyvalue_n])' indicates which policy value(s) the check should consider as acceptable:
 - Default is 'POLICY(WTOR,CANCEL,WAITNOH)'
 - The remaining possible policy value would be WAITHOLD
- INTERVAL is 24:00 (once a day), SEVERITY is LOW

Check details – Allocation

- (IBMALLOC, ALLOC_SPEC_WAIT_POLICY)
- Checks the value of the SPEC_WAIT policy (to be followed when an allocation request must wait for a specific volume or unit) in the current Allocation settings.
- This check ensure that the best SPEC_WAIT policy allocation setting for the particular environment is set. Certain SPEC_WAIT policy values can result in a deadlock.
- Check parameter 'POLICY(policyvalue_1[,...,policyvalue_n])' indicates which policy value(s) the check should consider as acceptable:
 - Default is 'POLICY(WTOR,CANCEL,WAITNOH)'
 - The remaining possible policy value would be WAITHOLD
- INTERVAL is 24:00 (once a day), SEVERITY is LOW

Check details – Allocation

- (IBMALLOC, ALLOC_TIOT_SIZE)
 - Checks the value of the TIOT (task I/O table) size in the current Allocation settings.
 - This check ensure that the best TIOT size value for the particular environment is set. Since the size of the TIOT is directly related to how many DD statements can be specified per job step, jobs may fail if there are more DDs than space for them in the TIOT.
 - Check parameter 'SIZE(min,max)' lets you specify a customized range for “acceptable” TIOT size values
 - Default is 'SIZE(24,32)'
 - INTERVAL is 24:00 (once a day), SEVERITY is LOW

Check details – Comm Server

- (IBMCS, CSRES_AUTOQ_GLOBALTCPIPDATA)
 - Checks, if the AUTOQUIESCE operand has been specified on the UNRESPONSIVETHRESHOLD resolver setup statement, but the GLOBALTCPIPDATA resolver setup statement has not been specified in the resolver setup file
 - Such a case would be flagged as check exception, since the AUTOQUIESCE operand would be ignored
 - INTERVAL is ONETIME, SEVERITY is LOW

Check details – Comm Server

- (IBMCS, CSRES_AUTOQ_RESOLVEVIA)
 - Checks if the RESOLVEVIA statement has been specified with the value TCP in the global TCPIP.DATA file when the autonomic quiescing of unresponsive name servers function is active.
 - The default value for the RESOLVEVIA statement is UDP. If the autonomic quiescing of unresponsive name server function is active, the resolver polls unresponsive name servers using UDP, not TCP. If your installation uses TCP, the results of the resolver's polling attempts will not accurately reflect the responsiveness of the name server in your installation.
 - INTERVAL is ONETIME, SEVERITY is LOW

Check details – Comm Server

- (IBMCS, CSRES_AUTOQ_TIMEOUT)
 - Checks if the configured resolver timeout value in the global TCPIP.DATA file exceeds the optimal setting when the autonomic quiescing of unresponsive name servers function is active.
 - The default value for the RESOLVERTIMEOUT statement is 5 seconds. If the autonomic quiescing of unresponsive name server function is active, the resolver polls unresponsive name servers every six seconds. The resolver uses the value you specify or default for the RESOLVERTIMEOUT statement in the global TCPIP.DATA file to determine how long to wait for a response to the poll. If you specify a value for the RESOLVERTIMEOUT statement that is greater than 5 seconds, the resolver will be less efficient when polling unresponsive name servers.
 - INTERVAL is ONETIME, SEVERITY is LOW

Check details – Consoles

- (IBM CNZ, CNZ_Console_Operating_Mode)
 - This check identifies installations running in 'Shared' console service operating mode.
 - 'Distributed' mode is the preferred mode of operations and 'Shared' mode will be removed in a future release.
 - INTERVAL is ONETIME, SEVERITY is MEDIUM

Check details – Consoles

- (IBM CNZ, ZOSMIGV1R13_CNZ_Cons_Oper_Mode)
 - The check identifies installations that haven't explicitly identified their console service operating mode.
 - For systems at a level prior to z/OS V1R13, the default console service operating mode is 'Shared', but beginning in z/OS 1.13 the new default is 'Distributed'. Simply accepting the default may leave the installation unaware that they are going to run in 'Distributed' mode.
 - INTERVAL is 24:00 (once a day), SEVERITY is LOW

Check details – Device manager

- (IBM DMO, DMO_TAPE_LIBRARY_INIT_ERRORS)
 - This check reports any tape library initialization errors that were detected during IPL.
 - Ensures that tape library HCD definitions agree with the tape library hardware definitions.
 - INTERVAL is ONETIME, SEVERITY is LOW

Check details – Supervisor

- (IBM SUP, SUP_HiperDispatchCPUConfig)
 - [Note: Has been available in R12, but was not presented due to related unannounced hardware]
 - This check monitors the number of CPUs installed and HiperDispatch state of the system. A warning is provided when HiperDispatch is disabled and the highest CPU ID of any CPU configured to this system is close to the maximum allowed.
 - A system with HiperDispatch disabled can use CPU ID 0 through CPU ID X'3F'. For the system to use CPU IDs above X'3F', the system must have HiperDispatch enabled.
 - Check parameter 'CpusLeftB4NeedHd(n)' specifies the minimum number of remaining CPUs which must be able to be installed and used with HiperDispatch disabled. Default is 8, the allowed range is 0 through 63.
 - INTERVAL is ONETIME (and for each HiperDispatch mode switch, or when a CPU is added to the system's configuration)
 - SEVERITY is LOW

Check details – zSeries file system

- (IBM ZFS, ZOSMIGV1R13_ZFS_FILESYS)
 - Verifies a Sysplex is running sysplex=filesys for zFS V1R13 toleration support.
 - zFS should be running at sysplex=filesys for all members of the sysplex. Once this is done, zFS V1R13 may be brought into the sysplex.
 - INTERVAL is ONETIME, SEVERITY is LOW

Check details – VSAM RLS

- (IBM VSAM RLS, VSAM RLS_QUIESCE_STATUS)
 - The check looks for unresponsive CICS® regions for QUIESCE and UNQUIESCE, which could indicate a problem.
 - Check parameter 'THRESHOLD(*elapsed_time*)' lets the check ignore rows in the quiesce status table with an elapsed time less than the specified number of seconds (0 through 2147483647). Default is 300.
 - INTERVAL is 00:05 (every 300sec), SEVERITY is MEDIUM

Check details – z/OS UNIX

- (IBMUSS, USS_PARMLIB)* updated
 - The check compares z/OS UNIX System Services' current system settings with those specified in the BPXPRMxx parmlib members used during initialization and issue an exception message if a difference is found.
 - This following additional z/OS UNIX dynamic settings will be checked by the updated check:
 - MAXUSERMOUNTSYS
 - MAXUSERMOUNTUSER
 - NONEMPTYMOUNTPT
 - LOSTMSG

Check details – Cross system coupling facility

- (IBMXCF, XCF_SFM_CFSTRHANGTIME)* updated
 - Monitors the CFSTRHANGTIME value in the Sysplex failure management (SFM) policy to make sure it is specified and it matches the check parameter CFSTRHANGTIME value. CFSTRHANGTIME specifies the amount of time before the system takes automatic action to relieve hangs caused when a connector fails to respond to structure-related events in a timely manner.
 - The default for the check parameter 'CFSTRHANGTIME(NO | *seconds*)', which specifies the wait time interval in seconds, has been changed from 300 (5 min) to 900 (15 min).
 - Installations must ensure that the check parameter reflects an appropriate value. Consider CFSTRHANGTIME carefully to arrive at a value that reflects the needs, goals, and prior experiences for a given configuration. In general, it is desirable to select a fairly aggressive value to limit sympathy sickness when connectors fail to respond to events in a timely manner, but this must be balanced against the possibility of terminating connectors prematurely.

Session summary

- The list of “regular” health checks and migration checks is growing, helping to keep your system healthy and easing the migration to new z/OS releases.

Appendix - References

- Related Publications
 - “IBM Health Checker for z/OS User's Guide” (SA22-7994)
 - Includes an inventory of all health checks known at publication time
 - See also http://www-03.ibm.com/systems/z/os/zos/hchecker/check_table.html
 - “Exploiting the Health Checker for z/OS infrastructure”
 - Redpaper 4590