IBM Education Assistence for z/OS V2R3

BCPii Hardware Support

Element/Component: z/OS BCPii

Agenda

- Trademarks
- Session Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Session Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Session Objectives

- BCPii performance improvements
- Absolute capping base support
- Absolute capping group support
- New more flexible HWICMD2 service
- Support for larger data returned from SE
- Support for dynamic CPC name change
- New HWISET2 service (multiple attribute set)

Overview – BCPii Performance Improvements

- Problem Statement / Need Addressed
  - BCPii query and list operations can be very slow
- Solution
  - BCPii's HWIQUERY and HWILIST APIs have dramatically improved performance
- Benefit / Value
  - Faster response times means happier customers!

Usage & Invocation – BCPii Performance

- Important post-z13 GA2 firmware update
  - BCPii Performance & Request Enhancements
    - MCL P00339.090 (Bundle S09)
- Significant performance improvements across the board for just about all kinds of BCPii requests and activity
- z/OS BCPii also made updates for performance improvements for activation profile queries

Interactions & Dependencies – BCPii Performance

- Software Dependencies
  - None
- Hardware Dependencies
  - MCL P00339.090 (Bundle S09)
- Exploiters
  - None

Migration & Coexistence Considerations – BCPii Performance

- None

Installation – BCPii Performance

- Rolled down to V2R1 and V2R2 via APAR OA51493.

Overview – Absolute Capping support

- Problem Statement / Need Addressed
  - New hardware features are not supported in BCPii
- Solution
  - BCPii fully the supports the querying and setting of all absolute capping attributes
- Benefit / Value
  - Users can more fully automate their control of the maximum consumption of resources a particular LPAR or group of LPARs can consume.

Usage & Invocation – Absolute Capping Base Support

•Allows to limit an LPAR’s maximum usage of any CPU type

- Requires target system to be at **zEC12 GA2** and higher

•New z/OS BCPii image and image activation profile attributes allow:

- An LPAR’s Absolute Capping enablement/disablement to be both queried and set for every processor type (GPP, IFL, IIP, etc..)
- An LPAR’s Absolute Capping value to the hundredeth’s of a processor to be both queried and set

Usage & Invocation – Absolute Capping LPAR group Support

- Allows to limit an LPAR group's maximum usage of any CPU type
  - Requires target system to be at **z13 GA2** and higher
- New z/OS BCPii image and image activation profile attributes allow:
  - An LPAR group's Absolute Capping enablement/disablement to be both queried and set for every processor type (GPP, IFL, IIP, etc..)
  - An LPAR group's Absolute Capping value to the hundredeth's of a processor to be both queried and set

Interactions & Dependencies – Absolute Capping

- Software Dependencies
  - None
- Hardware Dependencies
  - zEC12 GA2 for Absolute Capping LPAR (base) support
  - z13 GA2 for Absolute Capping LPAR group support
- Exploiters
  - None
-

Migration & Coexistence Considerations – Absolute Capping

- None

Installation – Absolute Capping

- Absolute capping base support
  - Rolled down to V2R1 and V2R2 via APAR OA49720
- Absolute capping LPAR group support
  - z/OS V2R3 only

Overview – HWICMD2 service

- Problem Statement / Need Addressed
  - BCPii's HWICMD service is not flexible enough to allow for additional command parameters or to increase the size of existing parameters.
    - e.g. BCPii's LOAD command only allows for 4-digit IPL addresses
- Solution
  - BCPii's new HWICMD2 service supports versioned parameter lists, allowing for future enhancements to specific commands to either accept additional parameters or to change their size.
- Benefit / Value
  - Immediate benefit: 5-digit IPL addresses for commands that take an IPL address when using the HWICMD2 service

Usage & Invocation – HWICMD2 service

- New HWICMD2 will use versioned parameter lists
  - Version passed into the service
  - Version 2 of the parmlist for Load, SCSI Load and SCSI Dump will have an extra byte to allow for 5-char IPL devices addresses
  - Eliminates the need for any foreseeable time in the future where the command service will need to be changed again

Usage & Invocation – HWICMD2 service

- The syntax of the new service in non-REXX will be as follows:

```
CALL HWICMD2(
ReturnCode,
ConnectToken,
CmdType,
CmdParm_Ptr,
CmdParmVersion,
DiagArea);
```

- The BCPii REXX interface for the command service will remain unchanged. BCPii will invoke HWICMD2, regardless if REXX program chooses Hwicmd or Hwicmd2

Interactions & Dependencies – HWICMD2 service

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration & Coexistence Considerations – HWICMD2 service

- None

Installation – HWICMD2 service

- Rolled down to V2R1 and V2R2 via APAR OA51250.

Overview – BCPii Constraint Relief

- Problem Statement / Need Addressed
  - Larger sized query attributes sometimes can exhaust BCPii buffers, resulting in a BCPii communications error
- Solution
  - BCPii increased the maximum sized data received by more than 10x.
- Benefit / Value
  - HWIQUERY will not fail in receiving attributes, no matter their size

Usage & Invocation – BCPii Constraint Relief

- New HWICMD2 will use versioned parameter lists
  - Version passed into the service
  - Version 2 of the parmlist for Load, SCSI Load and SCSI Dump will have an extra byte to allow for 5-char IPL devices addresses
  - Eliminates the need for any foreseeable time in the future where the command service will need to be changed again

Usage & Invocation – BCPii Constraint Relief

- Currently, some larger sized attributes can sometimes exhaust BCPii's max buffer allowed allowed
  - BCPii limits the maximum number of data chunks that can come back from the SE
- Verbose attributes (especially ones using XML) such as HWI_EC_MCL_INFO most susceptible to reaching the arbitrary BCPii limit.
- z/OS BCPii will now handle virtually any sized data returning from SE
- No application changes required

Interactions & Dependencies – BCPii Constraint Relief

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration & Coexistence Considerations – BCPii Constraint Relief

- None

Installation – BCPii Constraint Relief

- Rolled down to V2R1 and V2R2 via APAR OA50860.

Overview – Dynamic CPC name change

- Problem Statement / Need Addressed
  - An installation which dynamically changes the name of a CPC that has BCPii connections to it could have BCPii requests after the change sent to the wrong machine
- Solution
  - Fully support dynamic CPC name change
- Benefit / Value
  - Integrity of BCPii requests even during a catastrophic configuration change

Usage & Invocation – Dynamic CPC name change

- Today, z/OS BCPii does not react properly when an installation dynamically changes the name of their CPC and reboots the SE to make the change effective.
- BCPii has no easy way of knowing that a request targeting one CPC physical machine could suddenly now be arriving at another CPC physical machine. Example:
  - BCPii connections already established with CPCPROD
  - CPCPROD is renamed to CPCTEST
  - NewCPC is renamed to CPCPROD
  - Connections now flow to a new CPC box unknowingly!!!
    -

Usage & Invocation – Dynamic CPC name change

- BCPii will listen for when an SE is rebooted
  - The name of the CPC before and after will be compared. If local CPC name has changed:
    - All connections targeting the local CPC will be terminated
    - WhoAmI function on HWILIST will be updated to reflect the new name
    - BCPii will broadcast a Name Change ENF event to notify all parties on the local image that its name has changed
- BCPii will detect target of request has changed
  - Exploiting brand new updates in the firmware on zEC12 and z13
  - BCPii in conjunction with the firmware will detect if the target's CPC serial number has changed since the connection was started. If so:
    - The request will be terminated unsuccessfully
    - All connections to that target CPC will be terminated
    - BCPii will broadcast a name change ENF event to notify all parties on the local image that this CPC name has changed to a new name

Interactions & Dependencies – Dynamic CPC name change

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration & Coexistence Considerations – Dynamic CPC name change

- None

Installation – Dynamic CPC name change

- Rolled down to V2R1 and V2R2 via APAR OA51250.

Overview – HWISET2 service

- Problem Statement / Need Addressed
  - BCPii applications can only set one attribute at a time. Due to the higher latency connection between z/OS and the SE, if multiple attributes need to be set, there are time lags between each API call, resulting in a configuration that temporarily has incompatible values set for attributes.
- Solution
  - BCPii's new HWISET2 service leverages a recent hardware API enhancement, allowing for multiple attributes to be set on one call.
- Benefit / Value
  - Multiple attributes can be set at the same time, providing better adjustments to the hardware configuration, and resulting in shorter periods of configuration inconsistency.

Usage & Invocation – HWISET2 service

- New HWISET2 service ala HWIQUERY's retrieval of multiple attributes on a single call
- Additional benefit of having a "two-phase commit" like behavior where either all changes will be made or none of the changes will be made when targeting at z13 GA2 or higher machine.
- HWISET2 can target z13 GA1 and lower successfully.
  - If a failure occurs with one of the attributes, a return code of HWI_SET2_PARTIAL_UPDATE will be returned along with an indicator of the first failing attribute and the reason for the failure.
  - User will either have to manually back out the attributes already updated or attempt to rectify the partial update in another manner.

Usage & Invocation – HWISET2 service

- The syntax of the new service in non-REXX will be as follows:

```
CALL HWISET2(
                          ReturnCode,
                InConnectToken,
                SetParm_Ptr,
                NumOfAttributes,
                DiagArea);
```

`InConnectToken` must represent a BCPii CPC connection.

`SetParm_Ptr` points to an array of Set structures.

`NumofAttributes` specifies the # of elements in the array (up to 9 attributes can be set at one time)

Usage & Invocation – HWISET2 service

- Mapping of the SetParm structure pointed to by SetParm_Ptr:

```
/*------------------------------------------------------------------*/
/* Mapping of the SetParm structure, whose address is supplied by   */
/* as the SetParm_Ptr on a call to HWISET2.                         */
/* An array of these structures can be supplied to set multiple     */
/* attributes.                                                  @LGA*/
/*------------------------------------------------------------------*/
typedef struct ??<
    HWI_CONNTOKEN_TYPE Set2_CToken;
    int Set2_SetType;
    char *Set2_SetValue_Ptr;
    int Set2_SetValueLen;
??> HWI_SET2_SETPARM_TYPE;                                     /*@LGA*
```

Usage & Invocation – HWISET2 service

- New return codes:
  - HWI_SET2_SETPARM_INACCESSIBLE – setparm structure not addressable
  - HWI_SET2_NUMOFATTRIB_INV – specified a value <1 or >9
  - HWI_SET2_CONNECT_TOKEN_INV – connect token specified in the data structure either not associated with InConnectToken or bad value
  - HWI_SET2_PARTIAL_UPDATE – mentioned on previous slide
- New abend codes:
  - HWI_SET2_PRIM_ADDR – parms not in primary AS
  - HWI_SET2_NOT_ADDR – parms not addressable
  - HWI_SET2_BAD_PARM – number of parms bad

Usage & Invocation – HWISET2 service

- The BCPii REXX interface for the SET2 service will use constant stem variable tails ala HWIQUERY to set the various set values required.

```
stemName.0 = 3
stemName.1.SET2_Ctoken = <image1ConnToken on CPC1>
stemName.1.SET2_SetType = HWI_DEFCAP
stemName.1.SET2_SetValue = 5
stemName.2.SET2_Ctoken = <image2ConnToken on CPC1>
stemName.2.SET2_SetType = HWI_DEFCAP
stemName.2.SET2_SetValue = 3
stemName.3.SET2_Ctoken = <imageActProfConnToken on CPC1>
stemName.3.SET2_SetType = HWI_DEFCAP
stemName.3.SET2_SetValue = 5
```

- Note that the connect tokens must all be targeted to the same CPC as specified on the InConnectToken paramater.

Interactions & Dependencies – HWISET2 service

- Software Dependencies
  - None
- Hardware Dependencies
  - None
- Exploiters
  - None

Migration & Coexistence Considerations – HWISET2 service

- None

Installation – HWISET2 service

- None

Session Summary

- BCPii performance improvements
- Absolute capping base support
- Absolute capping group support
- New more flexible HWICMD2 service
- Support for larger data returned from SE
- Support for dynamic CPC name change
- New HWISET2 service (multiple attribute set)

.

Appendix

- All information will be documented in the ***z/OS MVS Programming: Callable Services for High-Level Languages*** publication.