SCP Diagnostic enhancements

Future - Trace log enhancements
PJ46160 - Diagnostic displays for name-value pair usage

Trace log enhancements

Michael Shershin

IBM Z

IBM

Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Problem Statement

When doing problem investigation, performance analysis, or enhancements, there is a need to understand the program path and resources that are used in z/TPF by a message. Existing tools can be used to get some information but key pieces of information are missing.

Pain Points

- When trace log is started, it does not collect information for child ECBs.
- When trace log is started, it only collects trace information for traces that are currently active.
- Trace log does not report when a copy-on-write happens.

To-Be: Child ECB processing

TLOGC START will have a new option to request a trace log session for all child ECBs

- A child ECBs is created when one of the following macros requests that an ECB be created: CREMC, CREDC, CREXC, CREEC, CRESC, SWISC TYPE=CREATE, fork(), tpf_fork(), pthread_create().
- Trace log will be automatically started for child ECBs as soon as the child ECB is created.
- Trace log will only be started for child ECBs that are created after the trace log session is started for the parent ECB.

As-Is: Traces used by trace log

When trace log is started, trace entries for all currently active traces are collected and reported.

For example:

- If C function trace is active, but extended C function trace is not active, trace log only reports function calls. Function exits are not included.
- If enter back trace is active, but TPFDF enter trace is not active, trace log does not report any TPFDF enters.

To-Be: Traces used by trace log

When trace log is started, trace entries for all possible ECB level traces are collected and reported.

- Traces will include: Macro trace with registers, extended C function trace, enter back trace, and TPFDF enter trace.
- If a trace is not active, it will remain not active for all ECBs that do not have a trace log session active. However, even if a trace is not active, it will be active for ECBs with a trace log session active.
  - For example, if C function trace is not active, it will be active for ECBs that have trace log active.

As-Is: Trace log controls

Trace log has two controls:

- Maximum number of trace log sessions allowed at one point in time.
  - ZASER TRLOG-x
- Maximum number of megabytes of storage allowed to be queued for a trace log session at one point in time.
  - ZASER TRLOGTH-meg

To-Be: Trace log controls

Existing trace log controls will be removed. Instead, resource checks for 1 MB frames will be made.

- Traces are collected and saved in 64-bit system heap. Today, a unit size of 4 KB is used for the system heap. A change will be made so that the unit size of 1 MB will be used.
- Preallocated 64-bit system heap buffers are used first. If memory beyond preallocated 64-bit system heap is needed, 1 MB frames are used. If too many 1 MB frames are used, trace log sessions will be stopped and the queued 1 MB frames will be returned.
- On a z15 collected trace log data will be compressed.

As-Is: Copy-on-write

When a copy-on-write happens, a counter for the number of copy-on-writes is incremented. However, there is no indication of what instruction caused the copy-on-write or what 4 KB page is being copied.

- If an ECB is taking a large number of copy-on-writes, it is difficult to determine where the copy-on-write is happening.

To-Be: Copy-on-write

A copy-on-write trace will be provided.

- When a copy-on-write happens, an entry will be put into macro trace. It will be flagged as C-O-W.
- Copy-on-write trace will be enabled or disabled via ZSTRC like all other traces.
- Copy-on-write trace can be used at any time. However, it will be most valuable with trace log because trace information is not lost when wrapping happens in the trace buffer.

To-Be: Other macro trace enhancements

Additional information will be included in macro trace

- ECB owner name: Include the new ECB owner name when it is changed
- CINFC: Include the type of CINFC (R, W, K) and the name of the tag.
- CORHC / ENQC: Indicate whether contention happened on the resource. If contention happened, how long did the ECB wait.
- Find macros: Indicate whether the record was retrieved from VFA or DASD.

Value Statement

When trace log is run, more complete information will be obtained. As a result a more thorough understanding of the path and resource usage for a z/TPF message will be gained.

PJ46160 - Diagnostic displays for name-value pair usage

Michael Shershin

IBM Z

IBM

Problem Statement

Name-value pairs can be used to characterize transactions. For example, what user made this request? Where does the user reside?

If a problem is happening in z/TPF, characteristics that are in name-value pairs can help understand the source and impact of the problem. Currently, there are no online diagnostic tools that can harvest the information in name-value pairs in real time.

Technical Details

- List the name of all name-value pairs that are currently in use

```
==> ZSTAT NVPLIST
CSMP0097I 10.27.41 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0104I 10.27.41 LIST OF THE IN-USE NAME-VALUE PAIR NAMES
                   COUNT OF THE IN-USE NAMES IS 4
                   search restrict
                   write restrict
                   ISrvcName       _
                   ISrvcVersion
END OF DISPLAY+
```

Technical Details

- Display counts of in use name-value pairs for a specified name

```
==> ZSTAT NVPNAME-ISrvcName
CSMP0097I 10.28.30 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0102I 10.28.30 LIST OF VALUES FOR THE ISrvcName NAME-VALUE PAIR NAME
                   IN-USE ECBS WITHOUT THE SPECIFIED NAME IS 349
                      COUNT       VALUE
                        24        tpffdesFileStat         _
                        10        addCard84wrapper
END OF DISPLAY+
```

Technical Details

- Display in use ECBs that have a specified name in a name-value pair.

```
==> ZDECB NVP 8 NVPNAME-ISrvcName
CSMP0097I 10.43.10 CPU-B SS-BSS  SSU-HPN  IS-01
DECB0026I 10.43.10 DISPLAY ECB SUMMARY USING NVPNAME-ISrvcName
ECB ADDR IS  PGM     TRC MIN SC VALUE
14F7B000  2 DJ00    DJ00 999  4 addCard84wrapper
14F8A000  1 DJ00    DJ00 999  4 addCard84wrapper
151CA000  1 DJ00    DJ00 999  4 addCard84wrapper
15359000  1 DJ00    DJ00 999  4 addCard84wrapper
16166000  2 DJ00    DJ00 999  4 addCard84wrapper
1647E000  1 DJ00    DJ00 999  4 addCard84wrapper
1657A000  1 CTHD    CTHD 999  7 tpffdesFileStat
16598000  1 CTHD    CTHD 999  6 tpffdesFileStat
TOTAL       34
END OF DISPLAY+
```

Technical Details

- Include count of in use name-value pairs for a specified name in output message when a no resources available dump is taken.
- No resources available dumps are:
  - 064C00, 064C01, 064C02, 064C03, 064C04, 064C05, and 0006DF
- Specify the name of the name-value pair to be used in the output message
  - ZCDNV NOSTORAGE ADD *name*

Technical Details

Example of a no resources available dump

```
CSMP0097I 21.49.06 CPU-D SS-BSS  SSU-HPN  IS-01
CPSE0161T 21.49.06 IS-0001 SS-BSS  SSU-HPN  SE-008720 CTL-I064C01 CATASTROPHIC
010012D TRC-QZZ5        OWNER-drvrQZZ5frmCVZZ-ECBAdr:17DF0000.
CP      OBJ-ccnucl          0000F22E
NO COMMON BLOCKS AVAILABLE

 LIST OF VALUES FOR THE  ISrvcName       NAME-VALUE PAIR NAME
 IN-USE ECBS WITHOUT THE SPECIFIED NAME IS 0000500.
       COUNT              VALUE
       0000024            tpffdesFileStat
       0000013            addCard84wrapper
       0000001            any1local

END OF NAME-VALUE DISPLAY
```

Value Statement

Understand the source and type of work that is currently running in z/TPF.

More quickly identify the cause of problems that are impacting your system.

➔ PJ46160 was available for download on November 9, 2020

Thank you

Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

Question	Answer
Copy-on-write is normal in C++ iostreams and strings. Does it contribute significant performance overhead ?	The overhead of a single copy-on-write is a little more expensive than a GETCC request but less than a find or file request. A small number of copy-on-writes per message does not cause significant overhead. However, if the number of copy-on-writes per message becomes large, the overhead might become noticeable. The new copy-on-write trace functionality will help identify where the copy-on-writes are occurring.

Summary of Q&A from the virtual TPFUG event:

Question	Answer
How are calls to fast link libraries handled?	An assumption is being made that the context of this question is concerned with how trace log handles calls to fast link libraries. The reason for this assumption is because the presentation is about trace log. A second assumption is that a fast link library is a collection of C functions. For tracing and trace log purposes, a C function in a fast link library is not treated any different than a C function in a standard shared object. A trace entry will show when the C function is entered (or called). A trace entry will show when the C function returns. The name of the shared object and the name of object and displacement within the object where the trace entry happened is provided. On entry, the C function parameters are provided. On return the return value is included.