Loaders Update

Operations and coverage

Josh Wisniewski

2025 TPF Users Group Conference
May 4-7, Austin, TX

IBM Z

IBM

PJ48028 (Jun 2024) Frequently entered program support for special linkage libraries

As-is

PJ45315 (2018) reduced overhead associated with linkage when frequently entered programs are in active loadsets. Special linkage libraries (CTAL, CISO, CTIS, CTAD, and CTDF) cannot be defined as frequently entered programs. While they are infrequently loaded, they can cause an increase in CPU utilization when loaded.

PJ48028 (Jun 2024) Frequently entered program support for special linkage libraries

To-be

- Special linkage libraries automatically use frequently entered program linkage to reduce linkage overhead when loaded to the z/TPF system with the E-type loader.
- To enable this support, just apply this APAR and the support automatically takes effect.

PJ48146 (Mar 2025)

Copy-on-write reduction for frequently entered programs.

Problem Statement

CPU utilization is impacted by copy-on-write processing.

- Programs that have global constructors and programs that have external references to data require “**first call processing**” – linkage code that must be run the first time each ECB enters the program.
- The system keeps track of whether an ECB has entered such a program by setting a flag in the program. This results in copy-on-write processing, which makes an ECB-unique copy of the flag.
- When multiple programs that require first call processing are called by most ECBs in your system the copy-on-write processing can increase processor utilization by a measurable amount.

User Story

Sophie
System Programmer

Sophie knows from looking at data reduction reports that her system has a high rate of copy-on-write processing. When she hears about PJ48146, she decides to investigate whether it can help reduce CPU utilization.

User Story

Sophie
System Programmer

Sophie loads PJ48146 to a native test system, starts a transaction simulator, and runs data collection. The enhanced data reduction report shows which programs cause copy-on-write processing because of first call processing.

User Story

Sophie
System Programmer

Using the information from the data reduction report, Sophie defines a number of programs as frequently entered programs in the program configuration file and loads it to her test system. She runs her test again and sees that the number of copy-on-writes has decreased. Even better, she sees a decrease in CPU utilization.

Background

Copy-on-write processing is caused by:

1. Applications that update static data
2. Updates to pointers to external data when new versions of programs are loaded
3. Setting the flag to indicate that first call processing has been done for a program with constructors or imported data <- These are the copy-on-writes targeted by this enhancement

Enhanced Data Collection Report Before Programs Are Defined As Frequently Entered

COPY-ON-WRITES BY PROGRAM MODULE BSS

PROGRAM	FEP	#FC_COW	FCCOW/sec	#C_O_W	C-O-W/sec	CWREL%	CWCUM%
MLIB	NO	11224198	14691.359	27034894	35385.987	14.492	14.492
YASK	NO	11321984	14819.351	25438944	33297.047	13.636	28.128
MLB2	NO	11437814	14970.961	23511432	30774.126	12.603	40.732
MLB3	NO	10749871	14070.512	21143216	27674.367	11.334	52.065
MAPP	NO	11364422	14874.898	11364422	14874.898	6.092	58.157
NOFC	NO	0	0.000	10953069	14336.478	5.871	64.029
YNOT	NO	10625268	13907.419	10625268	13907.419	5.696	69.724
NFC2	NO	0	0.000	10515794	13764.128	5.637	75.361
MAP2	NO	10309234	13493.762	10309234	13493.762	5.526	80.888
NFC3	NO	0	0.000	10153006	13289.250	5.442	86.330
YZEN	NO	9489732	12421.115	9489732	12421.115	5.087	91.417
TOTALS:		86522523	113249.377	170539011	223218.577		

Programs highlighted in yellow with >5000 first call copy-on-write/sec (FCCOW/sec) are all good candidates for frequently entered programs.

Enhanced Data Collection Report After Programs Are Defined As Frequently Entered

COPY-ON-WRITES BY PROGRAM MODULE BSS

PROGRAM	FEP	#FC_COW	FCCOW/sec	#C_O_W	C-O-W/sec	CWREL%	CWCUM%
MLIB	YES	0	0.000	18106727	23699.904	19.292	19.292
YASK	YES	0	0.000	16076960	21043.142	17.129	36.421
MLB2	YES	0	0.000	12799802	16753.668	13.638	50.059
NOFC	NO	0	0.000	10953069	14336.478	11.670	61.729
NFC2	NO	0	0.000	10515794	13764.128	11.204	72.933
NFC3	NO	0	0.000	10153006	13289.275	10.818	83.751
MLB3	YES	0	0.000	8694579	11380.339	9.264	93.015
TOTALS:		0	0.000	8729937	114266.934		

109K C-O-W/sec were eliminated and the overall C-O-W/sec was reduced by roughly 50% .
None of the top +90% of copy-on-writes are due to first call processing.

Value Statement

During testing, we saw a savings of approximately 225 million instructions per second (MIPS*) for each 50,000 copy-on-writes/sec that were eliminated.

*Large System Performance Reference (LSPR) standard used for measuring MIPS.

The actual savings you can expect to see is dependent on many factors.

Conclusion

PJ48146 reduces overhead associated with copy-on-write processing due specifically to first call processing that occurs when many ECBs call programs that contain constructors or that reference data in other programs.

This is most prevalent in applications written in C++.

Data collection and reduction are enhanced to help you determine how to benefit from this feature.

Thank you