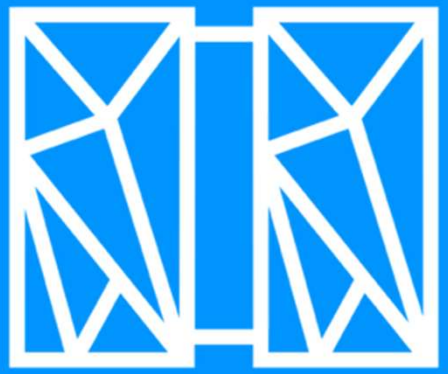
Recoverable Logical Record Cache

Chris Filachek
z/TPF Development



Disclaimer

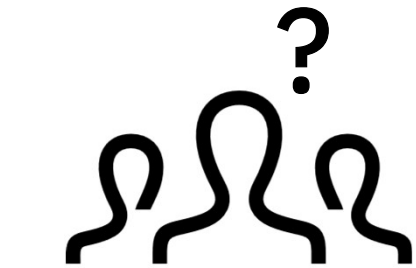
Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

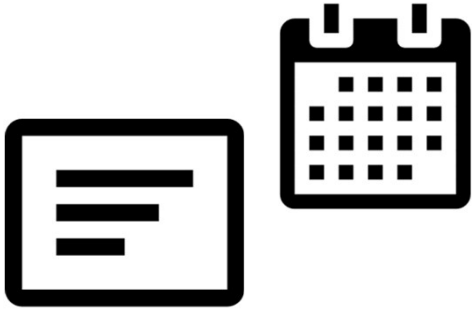
Background

- Logical record cache was introduced on TPF 4.1
  - Used by the z/TPF system
    - Cache file system information
    - Cache responses from DNS lookups
  - Cache the “answers” from I/O and processing – not just raw data
- Enhanced logical record cache provided on z/TPF in 2015
  - Supported cache entries larger than 4 KB
  - Allowed cache entries to be different sizes
  - Used by z/TPFDF to cache “mostly read-only” subfiles

Your Use Case for Caching?

- Do you have applications that repeatedly perform the same processing?
  - Availability? Schedules? Product details?
  - Are your customers asking you the same questions over and over again?
    - What flights go from NY to LA next Tuesday?
    - What amenities are in this hotel?
    - Do fries come with that?





Add Cache to Your Applications

- Cache “answers” from your applications
  - Can help save a significant amount of CPU and DASD I/O
  - Process once, reuse the “answer” many times

- To use logical record cache, update applications to…
  - Check the cache for the desired entry
    - If an entry is found, return the “answer” from cache
    - If not found:
      - Use existing logic to compute the “answer”
      - Store the “answer” in cache for future use

Lifespan of a Cache Entry

- Default: Cache entries stay in the cache until space is needed for new entries
  - Entries are reused based on a least recently used algorithm
  - Useful for data that does not change often

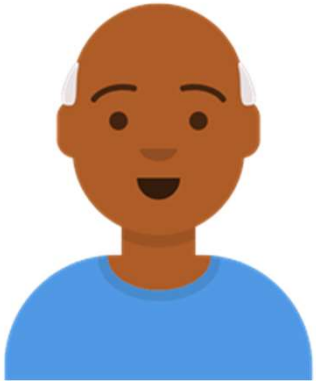
- Optional: Specify a cast out time (timeout)

  - Invalidates a cache entry after it is in the cache for N seconds
  - Can specify a different number of seconds for each entry
  - Useful for data that is valid only for some amount of time

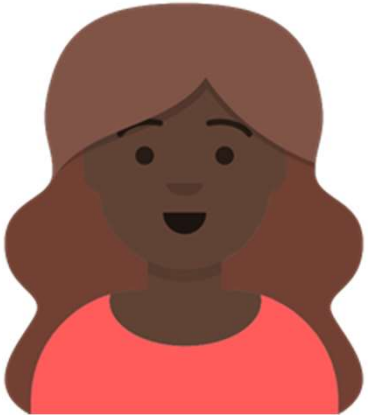
Problem

- All logical record caches and cache entries are lost across an IPL.
  - Applications can’t make effective use of the cache after an IPL because the cache is empty.
  - This might cause a period of high resource utilization (CPU, DASD, etc.) until the cache becomes sufficiently populated.
  - Depending on the size of a given cache, it could take a significant amount of time to naturally repopulate the cache.

Users



Derrick
operator



Carol
coverage programmer

Calvin
capacity planner

Pain Points

Derrick
operator

- Because the cache is lost across an IPL, we have to monitor system resources very closely after an IPL and watch for high resource utilization and input list shutdown conditions.
- If resources are low, we have to quickly adjust resources (add CPU), restrict certain types of traffic, stop or delay utilities, etc.

Pain Points

Carol
coverage programmer

- Restricting traffic can result in service disruptions to our customers.
- Some of our logical record caches are GB in size, so the impact (extra resources and service disruptions) could last several hours after an IPL.

Pain Points

Calvin
capacity planner

- To help mitigate the impact of an empty or ineffective cache after an IPL, we have activate extra resources like CPU.
- These extra resources are only needed immediately following an IPL and need to be deactivated after the cache is sufficiently populated.

Value Statement

- Allow the contents of logical record cache to be preserved across most IPLs such that the cache is effective as soon as z/TPF reaches NORM state.
  - Allow z/TPF applications that use logical record cache to recover to a steady state immediately following most planned and unplanned IPLs.
  - Preserve cache across most IPLs so the extra monitoring, resource usage, and service disruptions due to an empty cache can be avoided.

The Details

- Logical record cache is backed by system heap
  - System heap is set up from scratch during every IPL
  - All caches and cache entries are lost across all IPLs

- For critical caches, use recoverable system heap!
  - New type of system heap that is recovered across most IPLs
  - Applications that use logical record cache can choose between existing behavior or recoverable option

Using Recoverable Cache

1. Create a recoverable logical record cache: “MyRecoverableLRC”
   - Specify recoverable option to use recoverable system heap
2. Applications populate and use “MyRecoverableLRC”
3. The z/TPF system is IPLed
4. Create a recoverable logical record cache: “MyRecoverableLRC”
   - Existing cache is found by that name and automatically used
5. Applications continue using “MyRecoverableLRC” as if no IPL had taken place
   - z/TPF system quickly returns to steady state without needing to recreate or repopulate the cache

Recoverable across Most IPLs

- Cache should be recoverable across most IPLs if memory configuration has not changed
  - Soft IPL - ZRIPL and catastrophic system errors
  - Hard IPL - Same LPAR and without CLEAR
  - IPL with image change

- Cache is not recoverable across some IPLs
  - Memory configuration has changed
    - Different memory size or changed block allocations
  - Hard IPL: Different LPAR or IPL with CLEAR

Managing Logical Record Cache Options

Background

- Logical record cache options are set through APIs
  - Applications set options when creating the cache
  - Create the cache using newCache() or tpf_newCache_ext() APIs
  - Some of the options are:
    - Cache name
    - Cache size
    - Entry size
    - Entry count
    - Cast out time

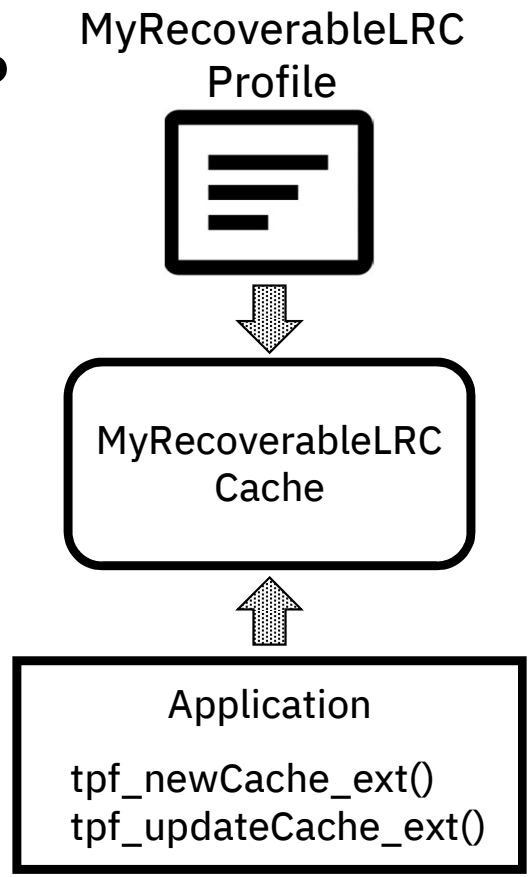
Pain Points

Calvin
capacity planner

- Because all logical record cache options are passed through APIs, we have to request application changes to update any cache options.
- Changing even a simple option like the cache size forces us to go through application code, build, test cycles, which makes it time consuming to try different options or make production changes.

Decouple options from Applications

- What if z/TPF supported “cache profiles” ?
  - Each profile contains the options for a logical record cache
    - Referenced by the cache name
    - Options could be changed without changing code
- Need input from Sponsor Users to determine details



Managing the Cache Size

Background

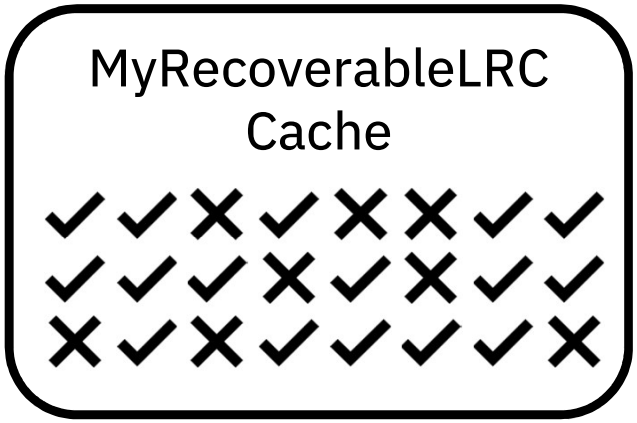
- If a cast out time (timeout) is set, cache entries are valid for a set period of time (seconds).

- Entries that have expired (timed out) are still in use and stay in the cache until space is needed for new entries.

Problem

- There is no mechanism to determine how many in-use entries are valid or how many have expired.

All cache entries are in use.



Only some cache entries are valid.

Pain Points

Calvin
capacity planner

- We need to make sure our logical record cache is as effective as possible, which usually means increasing the cache size to match our increased workload.
- Without knowing how many in-use entries are valid or expired, we can't easily determine the utilization of the current cache.

The Details

- Provide new metrics to help determine cache utilization that take expired entries into consideration.
  - Might include count of valid in-use entries or other metrics

- Might add metrics to a z/TPF command, the existing data collection metrics, or both.

Recap

- Preserve the contents of a logical record cache across most IPLs so z/TPF can recover faster to a steady operating state.
- Considering support for cache profiles so cache options can be managed outside of your application code.
- Investigating new metrics to help capacity planners understand cache utilization and size their caches.

Sponsor Users

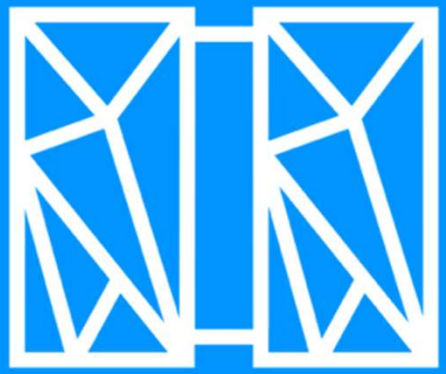
- Sponsor Users calls are starting soon to review our preliminary designs!
- Let us know you are interested so your needs are heard!
- Email drejza@us.ibm.com

Content Survey

ibm.biz/tpf-log-record-cache

Thank You!

Questions or Comments?



Trademarks

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Notes

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput 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 improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.