



**IDUG**  
2024 NA Db2 Tech Conference

## Application Modernization – Considered those Db2 for z/OS Capabilities?

**Maryela Weihrauch**

*IBM*



@IDUGdb2  
#IDUG\_NA24

Session Code: MODDEV3 | Platform: Db2 for z/OS

# Agenda

- Understand common pattern and data challenges in IBM Z application modernization
- Learn about Db2 capabilities and solution approaches for modern applications storing data in Db2 for z/OS
- Learn about best practices in using Db2 capabilities successfully

## Abstract

There are IBM Z application modernization initiatives ongoing in almost every organization using IBM Z Systems. Maryela shares common application modernization pattern and challenges. She will discuss Db2 for z/OS capabilities to support modern application architectures such as Microservices and share solution considerations and best practices.

Digital transformation imposes significant demands on existing applications and data

## Digital use cases drive data and functionality needs that are hosted on IBM zSystems



Account queries on Mobile using open APIs



Modernize loyalty program to hybrid cloud



Raise credit card dispute

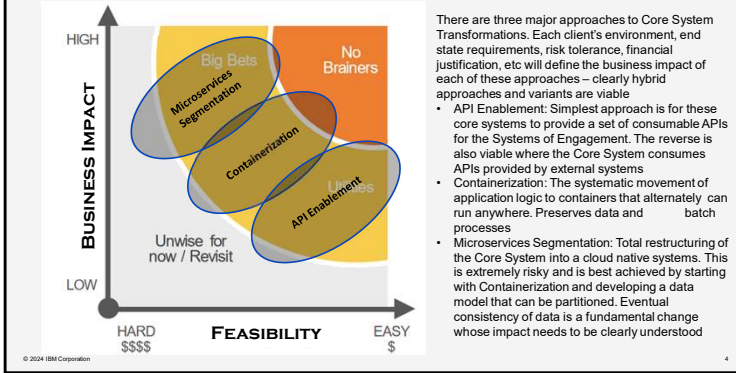
© 2024 IBM Corporation

Digital transformation is imperative for any business that is trying to delight their clients and effectively compete in the marketplace they serve. Any business that doesn't transform will be replaced by those that do – doesn't matter the business.

What is sometimes overlooked or less understood is the significant demands that digital transformation can have on existing applications and data. That's because timely access to data and data insight often fuels new business processes and client experiences. Take a simple example of a mobile banking application. Providing a real-time integrated view of a client's account information involves accessing multiple core banking applications along with derived account information and behaviors that might result in accessing various data and transaction systems. Many types of applications and data run (or originate) on IBM zSystems today, and often require some amount of integration and modernization to meet growing digital demands.

So, it's not a question of if you need to modernization, it's really a question of what's the best approach to achieve your modernization objectives.

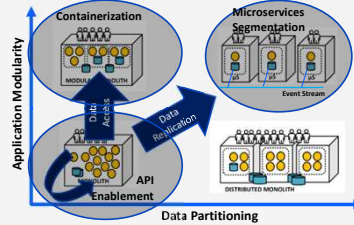
## Operational application transformation approaches



## Protecting/leveraging data will drive transformation approach

### Data Access vs Data Replication Considerations

- Establishing and maintaining a data replication pipeline is expensive and time consuming
- Maintaining a data replication pipeline typically creates data quality and data latency challenges for consuming applications
- Both Microservices and containerization enable cloud native, agile processes
- Accessing data in place accelerates the transformation, also improves opportunity for success
- Data access preserves the existing data management and recovery processes



### Data Consistency versus Eventual Consistency Considerations

- Data consistency reduces the amount of compensating application logic and/or business processes that need to be developed for systems that are out of sync
- Data Consistency addresses the double spend problem
- Eventual consistency enables systems to be developed independent of each other however can result in data quality issues

© 2024 IBM Corporation

5

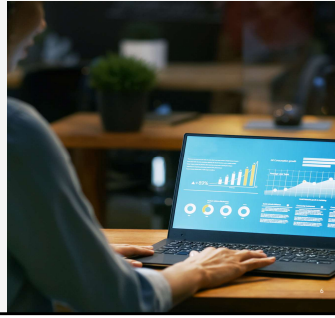
## Recommended incremental IBM Z data consumption modernization approach implements Data Fabric for IBM Z data

### **Communicate widely adopted incremental IBM Z data modernization approach**

1. Modernize access to existing IBM Z data
2. Virtualize and govern data access across data sources and platforms
3. Satisfy complex information needs (e.g., infuse AI models, historical data) within service processing

[IBM Architecture Center](#) for more information

© 2024 IBM Corporation



Modernize access to existing IBM Z data sources  
=> Prepare IBM Z data for integration into a Data Fabric

Enrich modern applications with real-time mainframe transactional data

Leverage broad API support via SQL and REST to simplify application development

Readily provide transformed data via SQL using the Db2 Analytics Accelerator and Data Virtualization Manager for z/OS

Reduce risk and increase data quality by reducing data movement



© 2024 IBM Corporation

7

Db2 is a relational database that can be accessed via SQL from anywhere

It supports native REST support to expose SQL and stored procedures as REST APIs

Other data sources can be accessed via SQL or REST api via Data Virtualization Manager for z/OS and z/OS

Connect EE

Where data is required in different formats, that is the consuming application needs the data in a specific format, you can readily transform the data's native format to that required by the consuming application

(using SQL syntax and accelerated by Db2 Analytics Accelerator). Data Transformation

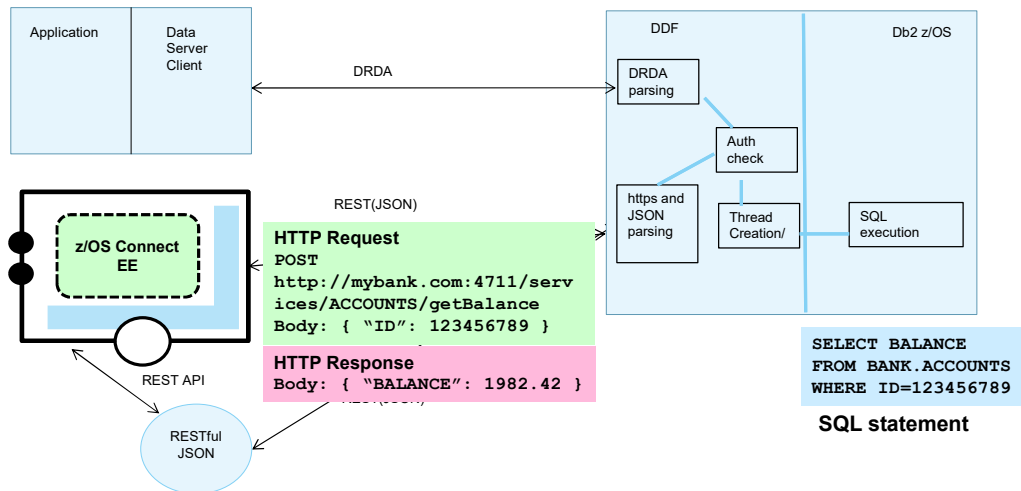
On Db2 Analytics Accelerator also mention temporal functionality is very relevant here to support as-of system time type queries and specific business application functionality related to changes in data over time (for example as a claim is submitted or as it is adjudicated).



## Db2 Native RESTful Service Support ...

Alternative distributed connectivity to Db2,

- No need for Db2 drivers, Db2 Connect gateway or Db2 Connect license
- Service call direct or via z/OS Connect



8

© 2024 IBM Corporation

Everyone is familiar with the existing Db2 distributed support, where a distributed application communicates with Db2 server via Db2 Connect or Data Server Driver for JDBC/SQLJ or CLI/.NET. The used communication protocol is DRDA.

With the RESTful service support, an application communicates with Db2 server using REST(JSON) as communication protocol. z/OS Connect would be used as service gateway for added functionality. Db2 Data Server Driver or Db2 Connect gateway is not required and therefore no Db2 Connect license is needed for this type of connectivity.

## Db2 RESTful Service Support

Db2 native RESTful service functionality is ready for use with Db2 installation  
Details in [Db2 for z/OS documentation](#)

- Reuses DDF functionality that provides a highly available, highly scalable and highly serviceable network interface
- DDF provides security, auditing, and data transformation as being provided using z/OS Connect
  - PERMIT DSN.REST CLASS(DSNR) ID(DB2USER) ACCESS(READ)
- Support of clientInfo fields
- Uses cloud and mobile application programming HTTP and HTTPS protocols
- zIIP eligibility of distributed workload
- Allows a database administrator to manage services
  - Catalog table SYSIBM.DSNSERVICE
  - Static execution pattern – auth checking, object dependency tracking

9

© 2024 IBM Corporation

There are a number of reasons for moving the RESTful service functionality into Db2 itself.

Before I describe them, I need to point out that the externals of the defined services using the Db2 Adapter for z/OS are not changed and client application will continue to work even the service definition is migrated to the native support.

The first big benefit, every Db2 sysprog will appreciate, the functionality is ready to use after the PTFs are applied and HOLD actions are done – something they are very familiar with.

The implementation builds on the Db2 DDF functionality that was optimized over the last couple of Db2 versions and provides a highly scalable, highly available network interface.

DDF provides security, auditing and very efficient data transformation. A new RACF permission REST is introduced to allow a user or group to access Db2 via HTTP/HTTPS.

The services are not managed in XML files but in a new catalog table

SYSIBM.DSNSERVICES in Db2, familiar to a DBA. Db2 uses a static execution pattern for authorization checking and can now do object dependency checking. That means if an SQL as an implementation of a service is not operational for any reason, the service is marked as not operational. That can be detected and corrected without calling a service.

# Application Modernization - Programming Languages on z/OS

Allows clients to:

- access a vast pool of developers to work on z/OS
- leverage a large number of “good-to-go” packages that can be used instantly in applications
- take advantage of the enormously popular programming languages to accelerate digital transformation and enable new applications on IBM Z
- co-locate applications and services on z/OS to improve performance in accessing critical applications and data

Support IBM Offerings and new technologies development

- Python: Red Hat Ansible Certified Content for IBM Z, ZOAU
- Node.js: Wazi, Zowe
- Go: Container runtime and Kubernetes orchestration for z/OS SOD: <https://ibm.co/3rAMDaB>

© 2024 IBM Corporation

IBM SDK for z/OS, Java Technology Edition (included in z/OS)

SDK for Node.js – z/OS\*

Open Enterprise SDK for Python\*

Open Enterprise SDK for Go\*

\*Available in ShopZ

- Zero license charge
- Optional priced S&S

Estimated number of developers Worldwide (2021)<sup>1</sup>

Java: 9.2M

Python: 8.4M

Node.js: 7.5M

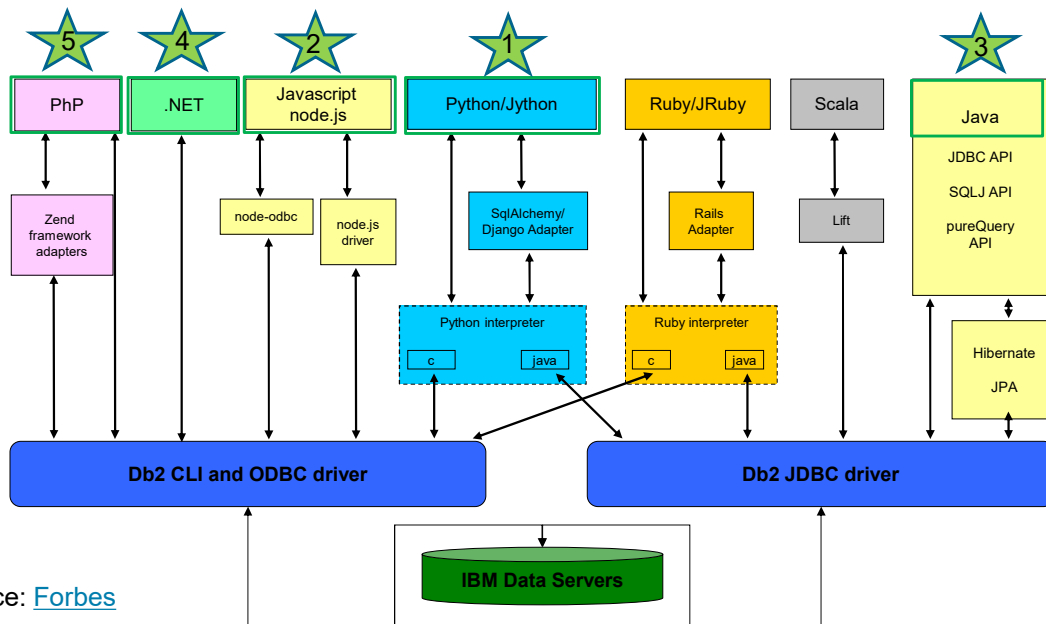
Go: 5.6M

<sup>1</sup> IDC – PasSView and the Developer 2020: Focus Topic Pivot Table — Worldwide and Regional

## Focus on staying current with open-source editions:

- Node.js: Shipped v12 in Dec. 2019 and v14 in Nov. 2020
- Python: Shipped first ever release, v3.8 in Jun. 2020; and Python v3.9 in Jan. 2021
- Go: Announced first release v1.16 in Jan. 2021; GA scheduled for Mar. 2021
- Java: SOD for Java 11 support <https://ibm.co/3jDctbm>

# Top-5 programming languages most in demand in 2023 ... can run on IBM zSystem and access Db2 for z/OS!



Source: [Forbes](https://www.forbes.com)

© 2024 IBM Corporation

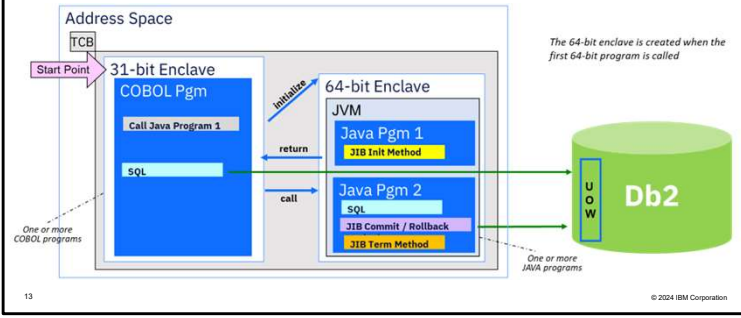
## Motivation for NEW Batch 31/64-bit LE/Java interoperability and Db2 Type 2 connection sharing support

- As part of many organizations' enterprise modernization strategy, Java is the preferred language to implement new or changed business functions.
- Because many are not starting from scratch, the new business function written in Java need to seamlessly integrate and interoperate with existing business logic written in COBOL or C.
  - **220+ billion lines of COBOL** - COBOL accounts for more than **70%** of the business transactions that take place in the world today<sup>1</sup>.
- A system programmer should be able to integrate the new application (existing application with Java extensions) into the existing and proven systems management framework (e.g. scheduling, workload management, security, monitoring, problem determination)
- Db2 z/OS Type 2 connection sharing provides single UoW across Cobol (LE) and Java modules
  - Alternatively, 2-phase commit needed between Cobol connection and Java connection to Db2 for z/OS
  - Already supported in WAS, CICS and IMS
  - New support for z/OS batch jobs

1 - <http://cobolpros.com/the-need-for-cobol/>

**Scenario example: 31-bit COBOL program → 64-bit Java program**

- COBOL program must call JIB Initialization method before making any Db2 calls.
- All Commits / Rollbacks need to be done by calling JIB Commit / Rollback Methods



13

© 2024 IBM Corporation

## Db2 Connection Sharing Support - Db2

| Description   | APAR    | PTF Availability                      |
|---|---------|---------------------------------------|
| <b>Db2</b> support that: <ul style="list-style-type: none"> <li>Introduces a new class to Data Server Driver for JDBC/SQLJ Type 2, specifically for Java Interlanguage Batch Db2 environment to support SQLJ</li> </ul> | PH48861 | November 15, 2022<br>(Db2 12, Db2 13) |

| Description  | APAR           | PTF Availability                          |
|--|----------------|---|
| <b>LE</b> support that: <ul style="list-style-type: none"> <li>Provides a new callable service to allow high-level language programs to query / set / unset the interoperability state of current task.</li> <li>Can also be used to query / set / unset if the current TCB is sharing a Db2 connection in the Java Interlanguage Batch environment</li> </ul> | <b>PH45216</b> | November 17, 2022<br>(z/OS V2.4 and V2.5) |
| <ul style="list-style-type: none"> <li><a href="#">Java Interlanguage Batch</a></li> </ul>   | PH58890        | 2024-03-27<br>(z/OS V2.4, V2.5 and V3.1)  |



## Need to load data from distributed systems into Db2 for z/OS

### ▪ Traditional solutions

- File transfer from client to z/OS and followed by Db2 LOAD utility
  - Invoke DSNUTILU stored procedure to call LOAD utility
- Other DRDA connection-based approaches
  - Client application can issue SQL INSERTs
  - Db2 Connect IMPORT utility with SQL INSERT option and IXF input
  - Db2 z/OS cross-loader from Db2 LUW tables

### ▪ Db2 12 for z/OS DRDA fast load

- Easy and fast loading of data from file that resides on client
  - **Java T4** application can use ZLOAD method in a Db2 connection class
  - Supported client file formats:  
Internal format, as well as delimited and spanned (LOB/XML data)
- Performance results showed DRDA fast load as fast as LOAD utility
  - Significant elapsed time reduction and up to 100% zIIP exploitation

# Db2 for z/OS DRDA Fast Load

## Data Server Client for JDBC/SQLJ

```

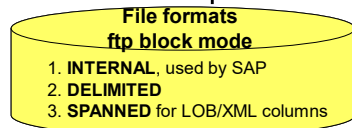
...
DB2Connection db2conn = (DB2Connection)con;

// LOAD statement text is in a string or input file
String loadstmt = "TEMPLATE SORTIN ...." +
"TEMPLATE SORTOUT ...." +
"SYSCLIEN WORKDDN(SORTIN) REPLACE PREFORMAT LOG(NO) REUSE
NOCOPYPEND FORMAT DELIMITED UNICODE INTO TABLE
MYID.CUSTOMER_DATA NUMRECS 30000";

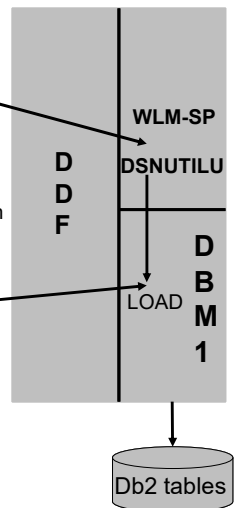
// Name of the file that contains the input data
String dataFilename = "C:\\customer.data";

// Identifier for this run of the LOAD utility
String utilid = "ZLOADTEST1";
LoadResult lr = db2conn.zLoad(loadstmt, dataFilename, utilid);

int returnCode = lr.getReturnCode();
String loadMessage = lr.getMessage();
    
```



## Db2 z/OS server



1. Connect
2. Establish LOAD environment
3. Start data stream and provide continuous blocks for LOAD

- Cancel of LOAD with RESUME YES  
 → RECP / RBDP, despite BACKOUT YES

The Db2 client ZLOAD enhancement allows to specify the full Db2 LOAD utility syntax.

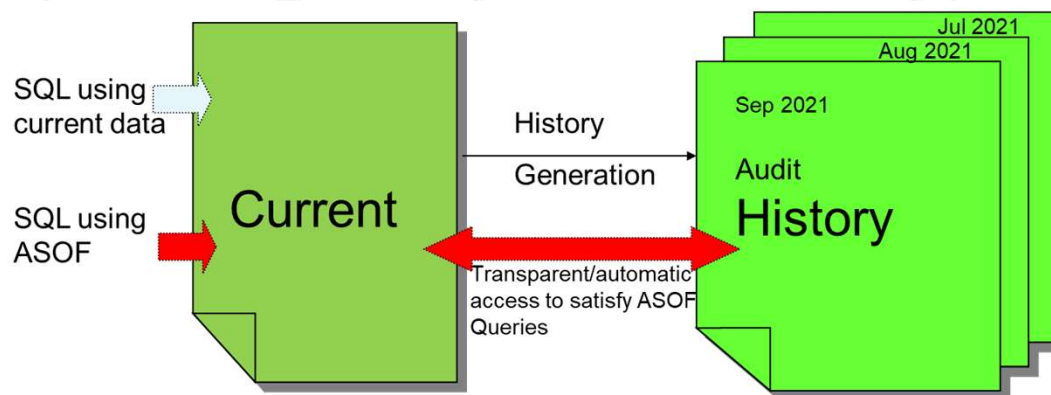
1. The Db2 CLP ZLOAD command is documented in IBM Knowledge Center:  
[https://www.ibm.com/support/knowledgecenter/en/SSEPGG\\_11.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0070172.html?view=embed](https://www.ibm.com/support/knowledgecenter/en/SSEPGG_11.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0070172.html?view=embed)
2. The Db2 CLI application enhancements will be documented in IBM Knowledge Center soon. The new CLI LOAD attributes, e.g. SQL\_ATTR\_Db2ZLOAD\_UTILITYID etc. were driven by SAP requirements.
3. A Java JDBC T4 driver application sample using ZLOAD method in a Db2 connection class is available on box folder topic\_optional:  
<https://ibm.ent.box.com/file/141072389922>

The data streaming is processed under the control of a system thread, which accounts the cpu time to the DIST address space and allows up to 100% zIIP offload.

## Need to manage different versions of application data

- Application programmers and database administrators struggle with managing different versions of application data.
- Regulatory laws require maintaining historical versions of data for years.
- Every update and delete of data requires old data be copied into history tables
- Existing approaches to application level versioning complicates table design and adds complexity and error prone code to applications

**Db2 Temporal - *SYSTEM\_TIME*:** History table contains version of every update on a single row



17

© 2024 IBM Corporation

### The Problem:

For years, application programmers and database administrators have been facing the problem of managing different versions of application data. With new regulatory and compliance laws requiring historical data to be maintained over years, the problem is becoming more acute requiring significant investment by DB2 customers in changing existing or creating new database applications to manage different versions of data. Every update or delete of data, the old data has to be copied into a history table. The exact table layout depends on the level of detail that is to be versioned. Data versioning can also be done via triggers.

### Impact:

Existing approaches to application level versioning not only explodes the table design but also adds complexity and error prone code to the application. The lack of data versioning in DB2 prevents the protection and management of core business sensitive assets by DB2.

### External aspects of a successful solution:

DB2 would provide a capability to specify table-level specifications to control the management of application data based upon time. Application programmers would be able to specify queries that specify a search criteria based upon time the data existed. This function would simplify and reduce the cost of developing DB2

applications requiring data versioning and allow customers to meet new compliance laws faster and cheaper because DB2 will automatically manage the different versions of data autonomically. The expectation is that select, insert and update of current data would perform similar for tables not performing data versioning, but select of old data may be slower.

# DB2 Temporal - *SYSTEM\_TIME*

## System-period Temporal Table (STT) contains *SYSTEM\_TIME* period

- ***SYSTEM\_TIME* begin column** can be defined as:

*TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN*

*TIMESTAMP(12) WITH TIME ZONE NOT NULL GENERATED ALWAYS AS ROW BEGIN*

- ***SYSTEM\_TIME* end column** can be defined as:

*TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END*

*TIMESTAMP(12) WITH TIME ZONE NOT NULL GENERATED ALWAYS AS ROW END*

- **Two table approach:** STT must have a history table
- **Inclusive-exclusive model:** for a row, the beginning value is included in the period and the end value is NOT included in the period
- **Query with FOR *SYSTEM\_TIME* period specification** (explicit, base table reference)

# Db2 Temporal - *BUSINESS\_TIME*

## Application-period Temporal Table (ATT) contains *BUSINESS\_TIME* period

- ***BUSINESS\_TIME* begin column** can be defined as:  
*TIMESTAMP(6) NOT NULL*  
*DATE NOT NULL*
- ***BUSINESS\_TIME* end column** can be defined as:  
*TIMESTAMP(6) NOT NULL*  
*DATE NOT NULL*
- **Inclusive-exclusive model** [...]: for a row, the beginning value is included in the period and the end value NOT included in the period  
or **Inclusive-inclusive model**: beginning and end value included in period (business time only)
- **Implicit check constraint** generated by DB2: `end_column > begin_column`
- **New index clause *BUSINESS\_TIME WITHOUT OVERLAPS***: enforce index keys being unique over a period of time
- **Temporal UPDATE/DELETE with *FOR PORTION OF BUSINESS\_TIME period clause*** (explicit, base table reference): modify rows for a portion of time

## Application modernization is an opportunity to create a solution for challenges of inactive data in Db2 tables

- A large percentage of data in a table can be inactive, results in (very) large tables
- Inactive data tends to be accessed less and less over time
  - One way to address: move inactive data to separate archive but not popular with business because of data accessibility from business functions
- Data maintenance (Image copy, reorg, runstats) requirements for active, frequently updated data is different than for inactive, static data
  - Inactive data can severely slow down recovery times during unplanned outages
- Large percentage of inactive, seldom accessed data impacts performance accessing active data
  - Lower BP hit ratio, larger index tree

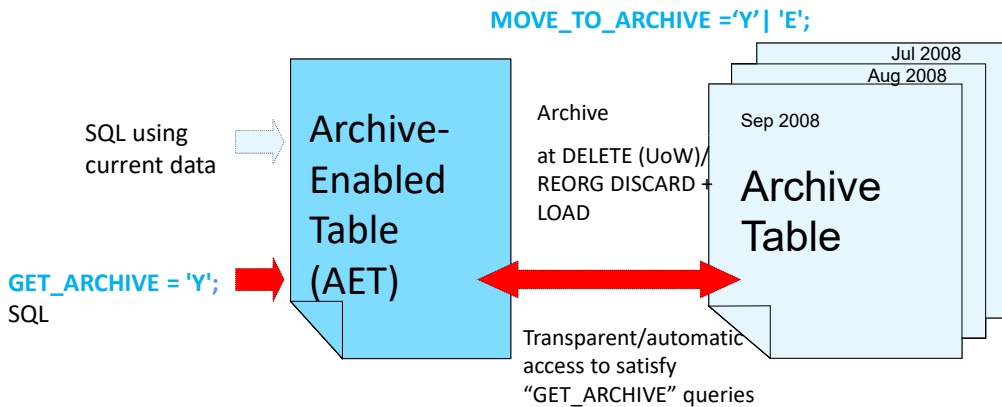




## Db2 Archive Transparency –

Separate inactive data while keeping access from existing application

Archive table contains rows deleted from archive-enabled table



Zparm MOVE\_TO\_ARCHIVE\_DEFAULT in macro DSN6SPRM

Gap (open requirement) Archive via REORG/DISCARD in single unit of work

21

© 2024 IBM Corporation

- Introduction of the built-in global variables -- the "buttons":

- `SYSIBMADM.MOVE_TO_ARCHIVE`: CHAR(1), value can be 'Y', 'E', or 'N', default 'N'

- `SYSIBMADM.GET_ARCHIVE`: CHAR(1), value can be either 'Y' or 'N', default 'N'

- Set the built-in global variable `SYSIBMADM.MOVE_TO_ARCHIVE` to 'Y' or 'E', the deleted records are propagated to archive table automatically by DB2 via single DELETE SQL statement. Executed within UoW.

- Set the built-in global variable `SYSIBMADM.GET_ARCHIVE` to 'Y' for all subsequent SQL statements including those from invoked function, stored procedure, and trigger. This allows the application to see both active and archive data without modifying the SQL statements in multiple packages. DB2 rewrites the query with UNION ALL operator.

- Archive via REORG/DISCARD + LOAD not done in single UoW – consider implications for data availability

## Db2 for z/OS: System profile monitoring

### ▪ Use cases:

- Prevent workload with denial-of-service attack characteristics
- Manage migration to new driver level
- Reserve threads for critical applications
- Set special registers or global variables
  - Including accelerator, transparent archiving enablement
- Non-distributed: anticipate production access paths in test
  - Model production values of RIDPOOL, SORTPOOL, BP allocations

### ▪ How to exploit for distributed clients:

- Define a distributed application to a profile by qualifier
  - E.g. IP address, or AUTHID and/or ROLE, or driver level, or ...
- Assign values to profiles
  - E.g. threads more granular than single DSNZPARM value for MAXDBAT
  - And/or set special register(s) or global variable(s) for profile(s)

22

© 2024 IBM Corporation

## Db2 System Profile Monitoring (SPM)

How to achieve these use case benefits:

1. cap connections or active threads for client profiles
2. Define profile by product id (PRDID) for out of support driver; start with WARNING, change to EXCEPTION when ready to enforce
3. Cap threads for one or more profiles to ensure threads remain for critical applications
4. Define profile for targeted apps or servers, use Attribute for details of SET command
5. Define profile in 'test' environment with no qualifiers. Attributes for that profile specify production values for RIDPOOL, SORTPOOL, BPs

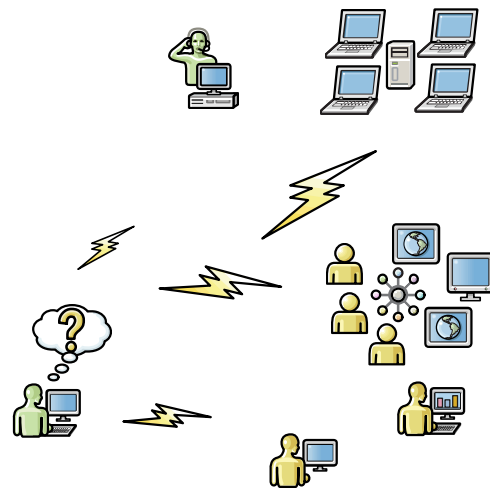
## Db2 for z/OS System Profile Monitoring

### ▪ Resources to manage

- Connections
- Threads
- Timeouts
- Special registers, Db2 defined global variables
- Db2 13 Profile table support for local threads

### ▪ Clients to support

- Priorities, demands
- Variety of applications



23

© 2024 IBM Corporation

There are many challenges facing the Db2 professional (System administrator, DBA, etc) when dealing with the variety of remote connections to Db2.

App servers with many different applications, executive users, analytics users, casual users.

Not all of these should have the same priority or access to resources. But there are limited options for control within Db2 definitions, such as single DSNZPARM values for maximum number of connections (CONDBAT), maximum number of active threads (MAXDBAT) and idle thread timeout (IDTHTOIN)

In the past, some customers used Db2 Connect Servers (aka gateway) as a way to manage resources and priorities for varying remote clients. Such Db2 Connect Server configurations presented several problems: for availability (additional point of failure), performance (additional code pathlength), and recovery (difficult problem determination). With Db2 system profile monitoring, Db2 Connect Servers should no longer be necessary in most cases.

## Summary

- Application modernization is a challenge and an opportunity
- Incremental modernization delivers value to the business fast with manageable risk
- Application modules implemented in different programming languages
  - All popular languages are supported with Db2 z/OS
  - Optimized 1-phase commit vs. 2-phase commit in Db2 z/OS Type 2 driver (OLTP, batch)
  - REST services (stateless)
- Efficient load of data from distributed platforms into Db2 with DRDA Fast Load
- Manage historical data needs with Db2 bi-temporal functionality
  - Also best performing approach for historical data in Data Warehouse and Data Lakehouse
- Manage inactive data with Db2 archive transparency
- As DBA use Db2 profile tables to influence application behavior without requiring application changes

24

© 2024 IBM Corporation

### Db2 System Profile Monitoring (SPM)

How to achieve these use case benefits:

1. cap connections or active threads for client profiles
2. Define profile by product id (PRDID) for out of support driver; start with WARNING, change to EXCEPTION when ready to enforce
3. Cap threads for one or more profiles to ensure threads remain for critical applications
4. Define profile for targeted apps or servers, use Attribute for details of SET command
5. Define profile in 'test' environment with no qualifiers. Attributes for that profile specify production values for RIDPOOL, SORTPOOL, BPs

## Notices and disclaimers

- © 2024 International Business Machines Corporation. All rights reserved.
- **This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.**
- Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM.
- Not all offerings are available in every country in which IBM operates.
- Any statements regarding IBM’s future direction, intent or product plans are subject to change or withdrawal without notice.
- IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks of International Business Machines Corporation, 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](http://www.ibm.com/legal/copytrade.shtml).
- Certain comments made in this presentation may be characterized as forward looking under the Private Securities Litigation Reform Act of 1995.
- Forward-looking statements are based on the company’s current assumptions regarding future business and financial performance. Those statements by their nature address matters that are uncertain to different degrees and involve a number of factors that could cause actual results to differ materially. Additional information concerning these factors is contained in the Company’s filings with the SEC.
- Copies are available from the SEC, from the IBM website, or from IBM Investor Relations.
- Any forward-looking statement made during this presentation speaks only as of the date on which it is made. The company assumes no obligation to update or revise any forward-looking statements except as required by law; these charts and the associated remarks and comments are integrally related and are intended to be presented and understood together.



**<Insert Session Title Here>**

**Speaker**

*Email Address*

*Session Code*



Please fill out your session evaluation!



**IDUG**

2024 NA Db2 Tech Conference

**Application Modernization –  
Considered those Db2 for z/OS  
Capabilities?**

**Maryela Weihrauch**

*[weihrau@us.ibm.com](mailto:weihrau@us.ibm.com)*

*Session Code MODDEV3*



Please fill out your session evaluation!



@IDUGdb2  
#IDUG\_NA24