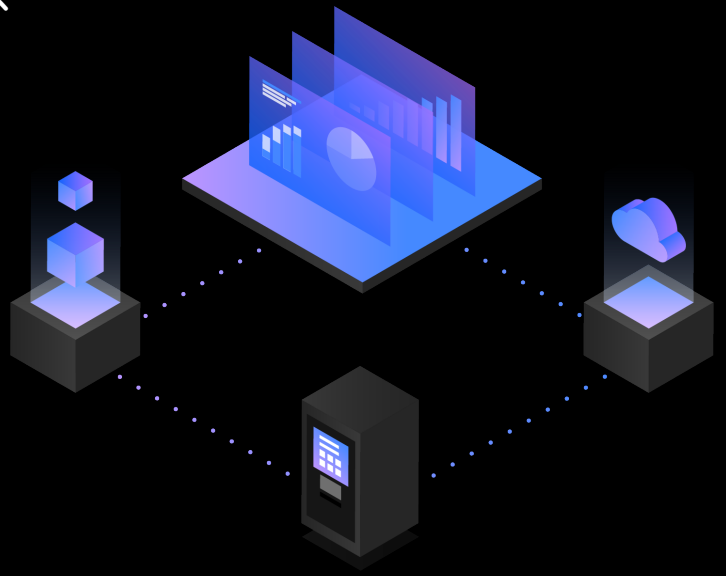
What Organizations are Doing (and looking to do) with Db2 for z/OS: Trends and Directions



Tridex Db2 User Group
May 20, 2021

Robert Catterall, IBM
Senior Consulting Db2 for z/OS Specialist

Agenda

- Interoperability – open data access interfaces
- Application development – support for DevOps processes, the rise of SQL PL
- Analytics – data gravity, machine learning
- Security – encryption at scale
- Availability and scalability – always online, massive vertical and horizontal scalability

Interoperability – open data access interfaces

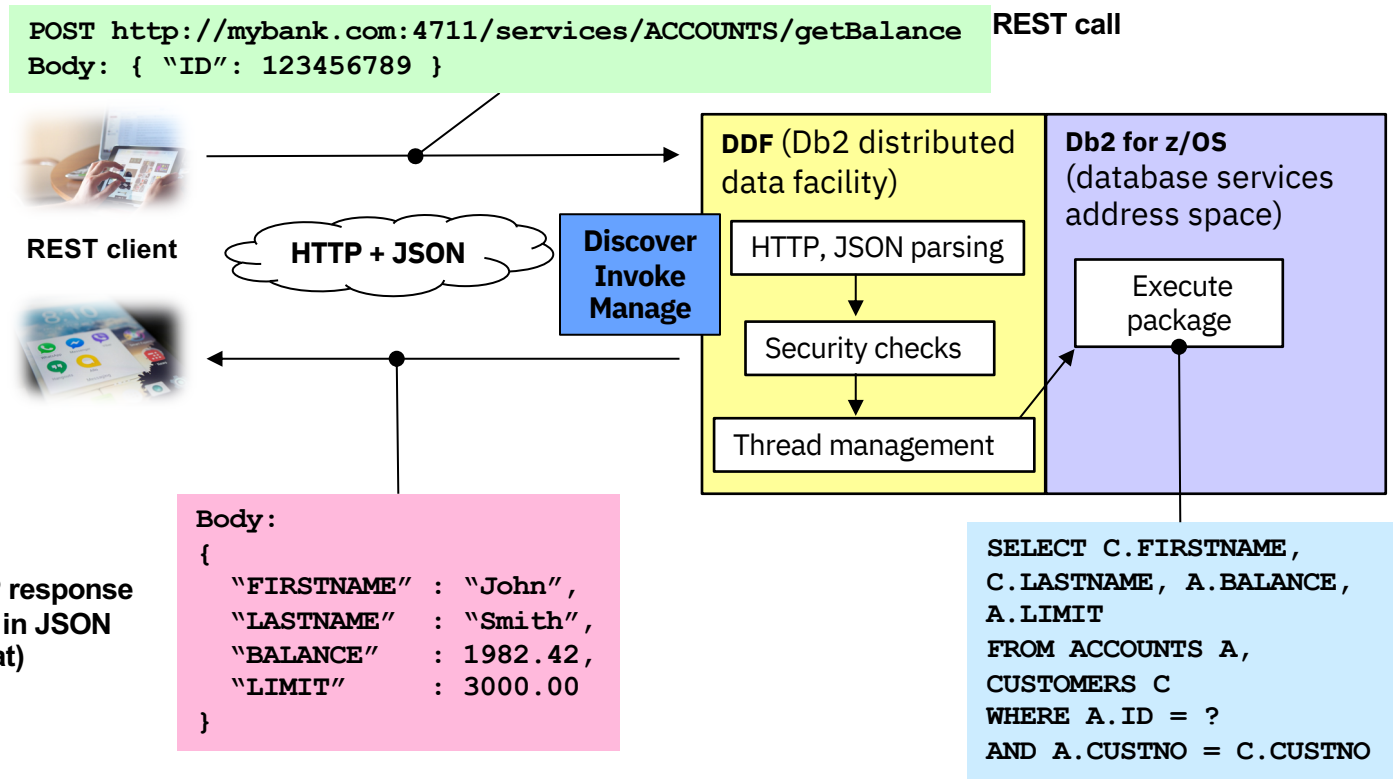
Db2 for z/OS open interfaces for SQL requests

- Db2 for z/OS has long provided support for “generic” SQL interfaces (i.e., SQL interfaces that are not specific to a particular relational DBMS)
  - Examples: JDBC (Java Database Connectivity), ODBC (Open Database Connectivity) and ADO.NET
  - Support for these interfaces for Db2 for z/OS access is provided by the IBM Data Server Driver (the successor to Db2 Connect) for network-attached applications
    - Local-to-Db2 applications can use JDBC/ODBC drivers provided by Db2
- Big help for developer productivity: no need for Db2-specific code – from client application perspective, Db2 for z/OS is just a relational data server

Data-as-a-service – the REST interface to Db2

- Designed initially for Db2 12 for z/OS, retrofitted to Db2 11
- As is true for SQL requests from network-attached applications, REST requests are handled via Db2 for z/OS distributed data facility (DDF)
  - Implication: up to 60% zIIP offload for SQL statement execution
- A single static (i.e., pre-coded and pre-compiled) SQL statement can be made invoke-able via a REST request
  - Statement could be SELECT, INSERT, UPDATE, DELETE, TRUNCATE or a CALL to a Db2 stored procedure
  - REST service can be created via a Db2-supplied REST service (called DB2ServiceManager) or via Db2 command BIND SERVICE command
- Same security protections as provided for SQL requests

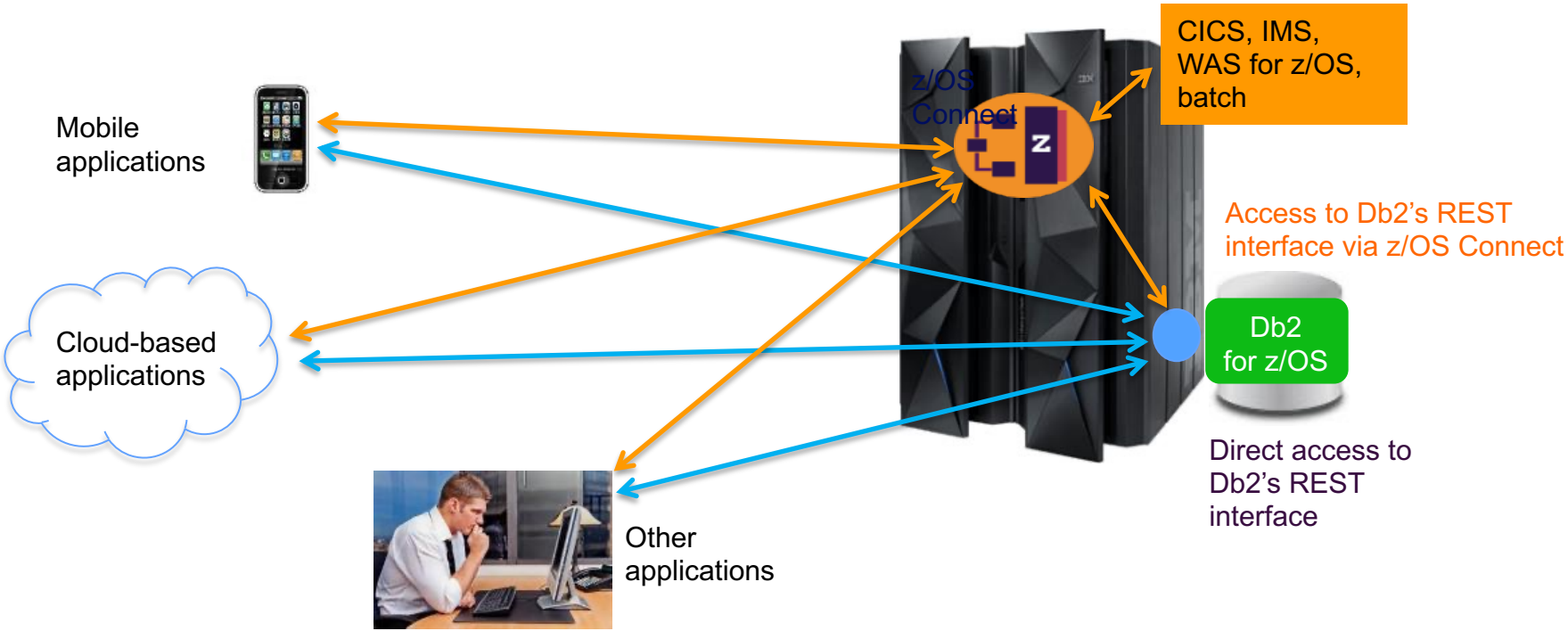
Flow of a REST request to Db2



Extending Db2 REST capabilities via z/OS Connect

- Application can access Db2 REST interface directly, or via z/OS Connect
  - In latter case, Db2 for z/OS is a *REST provider* to z/OS Connect
- Added value provided by z/OS Connect:
  - GUI tooling to facilitate creation of REST services
  - REST service description information in industry-standard Swagger format
  - Ability to use any HTTP verbs in REST call (e.g., GET and PUT)
    - Direct access to Db2's REST interface requires that request be in POST form
  - Greater flexibility in formatting JSON output documents
  - Single facility through which all manner of z/OS-based programmatic assets can be REST-enabled (e.g., CICS transactions and batch jobs, in addition to Db2 SQL statements)

Accessing Db2's REST interface – the big picture



Extending modern interfaces to data outside of Db2

- Formerly do-able via IBM InfoSphere Classic Federation Server for z/OS
- Newer, better way: IBM Data Virtualization Manager (aka DVM)
  - Better performance – federation performed in z/OS system rather than on an "in-between" server
  - Enables access to non-relational z/OS data (e.g., VSAM, Adabas, SMF trace data) via modern interfaces such as JDBC, ODBC and REST
    - Can be very helpful in migrating such data to Db2 for z/OS
  - Enables access to many data sources outside of z/OS, including DB2 for LUW, Oracle, SQL Server, but sources such as Hadoop, Hive, and mongoDB
    - Sources other than Db2 can be accessed as though they were DRDA servers (DRDA is Db2's distributed database protocol)

Open interfaces and Db2 for z/OS as enterprise data server

- Increasingly, Db2 used as data server for client-server applications
- More and more common to see Db2 for z/OS systems for which majority – sometimes large majority – of SQL statements executed via DDF
- Cost advantage: for SQL that gets to Db2 via DDF (or is issued by SQL PL stored procedures and functions invoked through DDF), up to 60% of execution can be offloaded to lower-cost zIIP engines
- Programmer productivity advantage: when using generic SQL interfaces or REST interface, developers do not need to know particulars of Db2 for z/OS
  - This is why vendor-supplied applications (e.g., SAP, Oracle PeopleSoft) that support Db2 for z/OS are typically written to access Db2 via DDF

Application development – support for DevOps processes, the rise of SQL PL

Going beyond industry-standard interfaces

- For developers working with Db2 for z/OS, non-Db2-specific interfaces (e.g., JDBC, ODBC, REST) are good for productivity, but more is needed
  - Talking about Db2 for z/OS database-as-a-service: automation and developer self-service in development environments
    - Example: enable developer to provision a schema (set of tables and related database objects), with associated data, and to modify tables in the schema, without having to always directly engage a Db2 DBA
  - Objective: enable developers to move faster when creating Db2-based applications
- Db2 for z/OS database-as-a-service can be implemented via an IBM offering called Db2 DevOps Experience for z/OS (DOE)

Db2 DevOps Experience for z/OS – what it does

- With Db2 DevOps Experience, Db2 team can create one or more “Db2 DevOps zones” in development systems within which developers can…

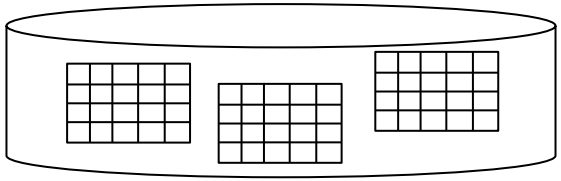
  …provision and manage application database schemas (set of tables) that are populated with data

  …alter the definition of objects in the schema

  …de-provision an application database schema

  …request changes to development team’s master schema

Developer makes Db2 database service request, and it gets done



Developer self-service in environment managed by Db2 team

- Through Db2 DevOps Experience, Db2 for z/OS team can…

  …define teams of users, environments, and database components for developers to use in their application development activities

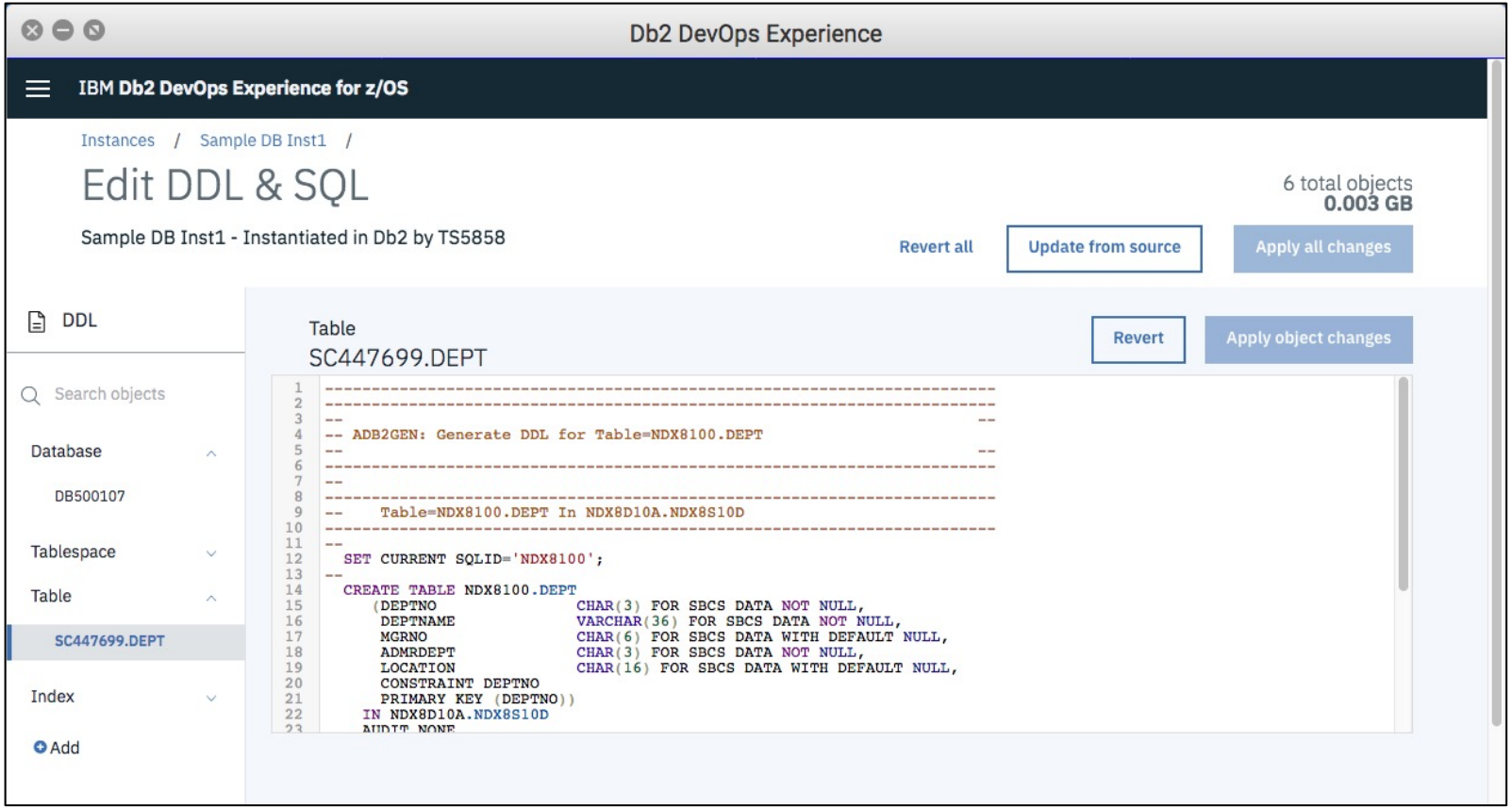
  …discover and select application's database components to be managed together

  …set storage limits and rules for schema and database names for defined teams

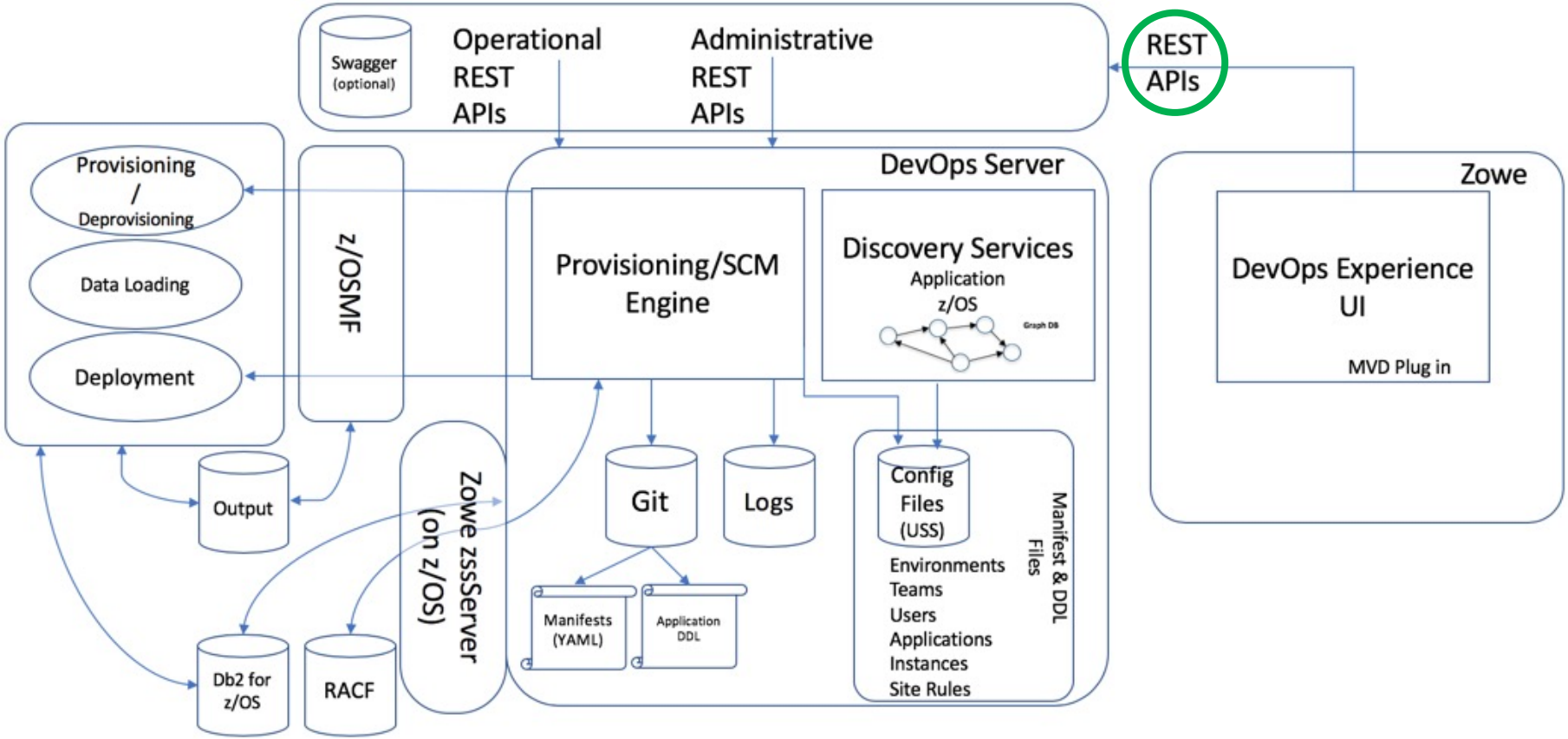
  …review and approve developer schema changes

Application developers get freedom to operate within a Db2 environment that is defined and managed by the Db2 team

Example of UI: DDL editor enables developer to view or modify definition of an object in a schema instance (note: in addition to built-in UI, DOE services can be invoked from DevOps tools via REST requests)



Db2 DevOps Experience for z/OS - architecture



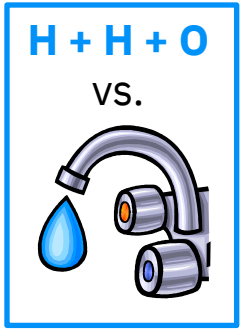
Trend: growing use of SQL Procedure Language

- SQL PL enables Db2 routines (stored procedures, user-defined functions, advanced triggers) to be written entirely in SQL
- Cost advantage versus Db2 routines written in languages such as COBOL: when invoked through DDF – whether via SQL CALL or REST request – execution of SQL PL routine is up to 60% zIIP offload-able
- Functional benefit: some capabilities only available with SQL PL routines
  - Example: autonomous stored procedures (data changes made by such a procedure are preserved, even if calling transaction subsequently rolled back)
  - Example: input can be Db2 array (especially appreciated by Java programmers)
- Another plus: SQL PL virtually identical between Db2 for z/OS, Db2 for Linux/UNIX/Windows

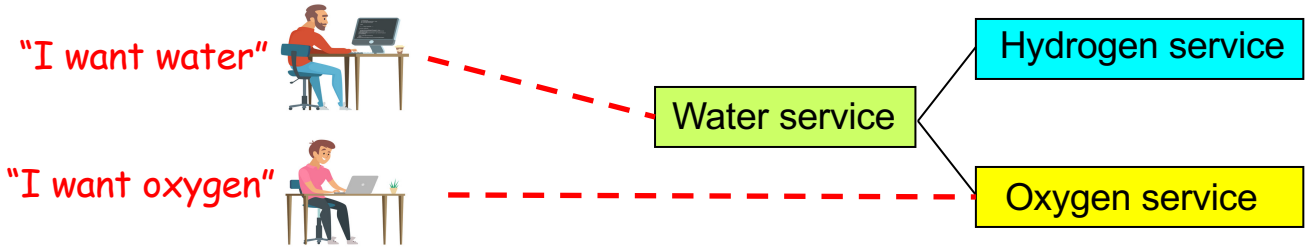
SQL PL routines and z/OS tasks

- An “external” Db2 routine, such as a stored procedure written in a language other than SQL PL, always runs under its own TCB
  - Db2 thread of invoking process must be switched to task of external routine
- A SQL PL routine never has its own task – it is simply a Db2 package that runs under the task of its invoker
  - No need to switch invoker’s Db2 thread to another task
- “No additional task” aspect of SQL PL routines particularly advantageous when “nested” stored procedures are used (i.e., when one stored procedure calls another)
  - And nested stored procedures can be a very good way to manage service granularity…

Service granularity and nested Db2 stored procedures

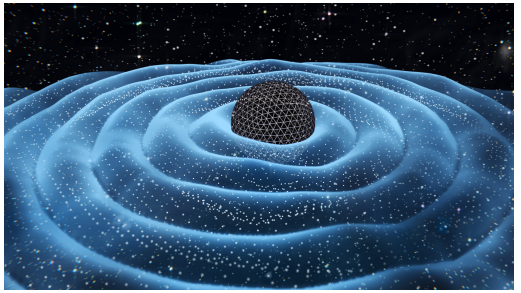


- Microservices (fine-grained services) can be great for flexibility, but can mean extra work for application developers
  - Developer: “I have to ask for two atoms of hydrogen and one atom of oxygen when what I want is water”
- Using nested Db2 stored procedures, fine-grained data services can be directly invoked when desired, or combined “under the covers” to provide more coarse-grained functionality
  - And with native SQL procedures, nesting does not involve additional z/OS tasks



Analytics – data gravity, machine learning

Data gravity



- The idea: leverage data where it lives
- In an analytics context, that means bringing analytics to the data, versus moving data to other platforms for analysis
- Db2 12: query optimizer enhanced to improve performance of complex queries (those involving outer joins, user-defined functions, UNIONs, etc.)
  - Some queries consume 90% less CPU when optimized by Db2 12 versus Db2 11
- Additionally, major enhancements to Db2 Analytics Accelerator with V7.5
  - The Accelerator is an analytics-optimized Db2 system that is tightly coupled with a “front-end” Db2 for z/OS system
  - Accelerator is logically invisible – query directed to front-end Db2 system, and that system routes query to Accelerator if doing so would reduce execution time

Db2 Analytics Accelerator V7.5 – key enhancements

- New deployment option: “virtual” Accelerator runs in containerized form on IFL engines in same Z server as front-end Db2 system
- Leverages Db2 BLU Acceleration technology
  - In-memory, column-oriented data structures, optimized for analytical queries
- New data-change replication mechanism keeps copy of table on Accelerator within 1-2 seconds of currency relative to table on front-end Db2
- New “pass-through” capability supports analytical functions that are built into Db2 with BLU but are not provided by Db2 for z/OS (e.g., LEAD, LAG)
  - Effectively makes those functions available for front-end Db2 for z/OS system

Letting Db2 for z/OS take advantage of machine learning

- IBM Db2 AI for z/OS (Db2ZAI, for short) puts machine learning to work in a Db2 for z/OS system
  - Utilizes IBM Watson Machine Learning for z/OS, but no data scientist support needed – Db2ZAI handles development, training, deployment and monitoring of predictive models

Db2ZAI: three main areas of functionality

- Enhanced query **re**-optimization: Db2ZAI learns about a query workload
  - Substitution values for parameter markers / host variables in predicates, number of result set rows fetched
  - Queries can be re-optimized using predictive models based on that knowledge
- System assessment: Db2ZAI learns about system's operational profile
  - Can then – on scheduled basis or on-demand – assesses system's activity
  - Anomalies are identified, mitigating actions recommended
- Distributed connection control: Db2ZAI learns about system's DDF activity
  - Recommends connection, thread limits for client systems (implemented via Db2 profile tables)
  - Goal: keep lower-priority DDF-using applications from monopolizing a system's connection and DBAT capacity

Security – encryption at scale

z/OS data set encryption

- A z/OS feature for encrypting data “at rest” (i.e., on disk)
- Application transparent – data encrypted when written to disk, decrypted when read into memory
- Based on association of an encryption key label with a data set
  - Access to decrypted data requires that user ID be permitted to access data set’s key label – permission provided by z/OS security manager (RACF or equivalent)
    - Note: from z/OS perspective, ID that accesses Db2 data is Db2’s ID
- CPU overhead for z/OS data set encryption relatively low for z13 server, very low for z14 and z15 (often 1-2%)
  - Very feasible to encrypt huge number of Db2 for z/OS data sets

Db2 12 makes z/OS data set encryption easier to implement

- Db2 12 *function level* 502 delivered significant usability enhancement:
  - DBA can ALTER a table (or a Db2 STORGROUP – a collection of database objects) with a KEY LABEL specification
    - Next time online REORG is executed for table space, data in table (and in associated objects such as indexes) will be encrypted on disk
    - New ZPARM parameter ENCRYPTION_KEYLABEL: key label for encrypting data in catalog and directory (encryption accomplished via online REORG), and archive log disk data sets (encrypted when created)
- Prior to *activation* of Db2 12 function level 502, associating an encryption key label with a Db2 data set has to be done outside of Db2:
  - RACF (or equivalent) data set profile, IDCAMS (at data set create time), or SMS data class specification

Db2 12: new built-in function for column-level encryption

- ENCRYPT_DATAKEY available with activation of Db2 12 function level 505
- Data encrypted in memory as well as on disk
- More-robust functionality versus predecessor function ENCRYPT_TDES
  - 256-bit encryption versus 128-bit
  - Ability to make encrypted form of duplicate values in a column distinct
  - To read decrypted column values, application uses DECRYPT_DATAKEY function, *and application's primary authorization ID requires access to key label used to encrypt data* (access provided by RACF or equivalent)
    - With ENCRYPT_TDES function, access to encrypted data permitted via specification of password used when data encrypted

Encryption of Db2 for z/OS data “on the wire”

- Db2 for z/OS has long supported use of SSL encryption for client-server communications (also known as AT/TLS encryption)
  - A Db2 system has a “standard” SQL listener port and can optionally have a secure SQL listener port
  - When client accesses system via secure port, Db2 requires use of SSL encryption
    - SSL “handshake” involves client presenting certificate provided by z/OS host
    - Also required: AT/TLS policy on z/OS side
  - Recent Db2 12 enhancement: a Db2 system’s *one and only SQL listener port* can be a secure port (APAR PH08188)
    - Any network-attached client application will be required to use SSL encryption

Availability and scalability – always online, massive vertical and horizontal scalability

Availability – reduce need for application data unavailability

- An ongoing imperative since Db2 for z/OS first came to market
- Some Db2 12 for z/OS enhancements in this area:
  - Insert new partition in middle of range-partitioned table with ALTER + online REORG
  - Allow concurrent read access to table while LOAD utility with REPLACE is executing
  - New SQL statement TRANSFER OWNERSHIP: change OWNER of database objects from one ID to another without having to unload/drop/re-create/reload
  - Rebind phase-in: successfully rebind a Db2 package even while that package is being executed by applications

Vertical scalability: do more with one…

- More power for one Db2 subsystem:
  - One z15 can have 190 processors, each with more than 1000 MIPS of capacity
    - Z14: 170; z13: 141
  - One z/OS system can address up to 4 TB of memory (i.e., “real storage”)
    - Db2 12: up to 16 TB of buffer pool space (ready when z/OS can address > 4 TB of memory)
- More data in one table with Db2 12 RPN (relative page numbering – a new type of universal partition-by-range table space)
  - Up 280 trillion rows
  - Up to 4000 TB of data

Data sharing: taking scalability, availability to the next level

- Horizontal scalability: up to 32 Db2 subsystems co-own one copy of data
  - Near-linear increase in processing capacity as members added to Db2 group
- Ultra-high availability:
  - Virtually eliminates need for planned outages:
    - No need for outage to apply software maintenance or change server hardware
    - Can even migrate from one version of Db2 to another *without ever stopping the application workload* – done for years, by a growing number of organizations
  - Greatly reduces scope of an unplanned outage:
    - If member fails, only pages/rows in midst of being changed at time of failure unavailable – automatic restart of failed member restores access to that data
- Invisible to applications – appears as single-image Db2 system

In summary…

- Db2 for z/OS aims to support both high-volume transactional and complex, data-intensive analytical processing on the same platform, with unrivaled scalability, availability, and security, in a way that makes the particulars of the platform invisible to application developers

Robert Catterall

rfcatter@us.ibm.com