

DB2 Connect Configurations

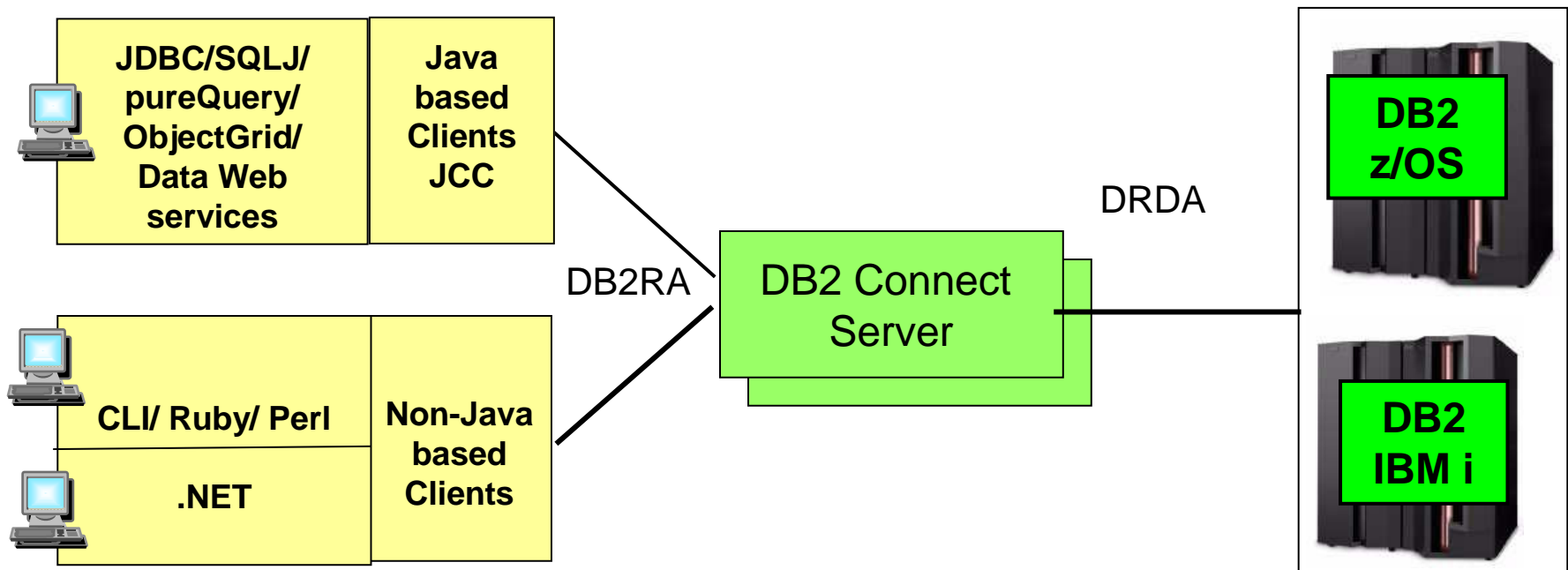


Brent Gross - gross@ca.ibm.com

Agenda

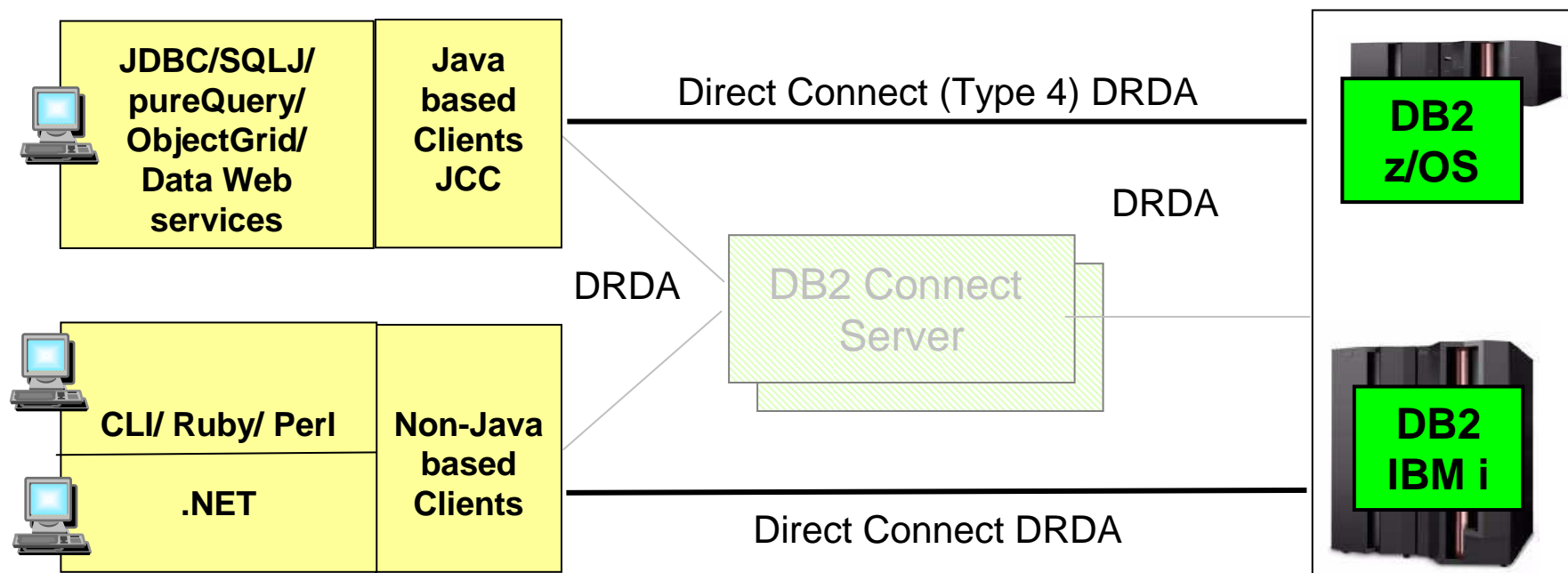
- **Review DB2 Connect configuration**
- **Where do I want to be**
- **Why should I go there**
- **Why would I still use DB2 Connect Server**
- **Mechanics of getting there**
- **Sysplex Workload Balancing**
- **ODBC and .NET function Rollout**
- **DB2 Connect Advanced Edition - pureQuery**

DB2 Connect Client Configuration – History



- Prior to DB2 Connect V8, DB2 Connect converted DB2RA (protocol for DB2 LUW) into DRDA (protocol for DB2 for z/OS, DB2 for IBM i)
- Starting with DB2 Connect V8, Clients use DRDA protocol natively and have the ability to connect directly to DB2 for z/OS and DB2 for IBM i

DB2 Connect Client Configuration – Recently



- Clients using DRDA directly allows DB2 Connect Server to be removed
- Clients - especially application server - access DB2 for z/OS directly
- Sysplex Workload Balancing was supported by JCC type 4 since V8 and by non-Java since 9.5 FP3

Where Do I Want to Be?

- **Lightweight client packages utilized throughout my enterprise**
 - IBM Data Server Driver Package (ds driver - recommended)
 - Staying current with driver levels much easier to attain

- **Direct connection to my DB2 for z/OS or IBM i server**
 - No DB2 Connect Server to maintain

- **Server based DB2 Connect license key**
 - One time install to server, no need for license key on client workstations
 - DB2 Connect Unlimited Edition for z/OS

Why Do I Want to Get There – Driver?

- **Recommended client package is ds driver**
- **Latest code levels required for new server feature exploitation**
 - For example, DB2 9.7 FP3a drivers required for new z/OS V10 datatypes
- **Some new features requiring data transport**
 - For example, new datatypes would require a DB2 Connect Server, if present, to also be at DB2 9.7 FP3a
- **Latest code levels required for API specific features**
 - For example, .NET 4.0 or Visual Studio 2010 require DB2 9.7 FP4
- **Larger client packages require more effort to upgrade**
 - Many customers still at DB2 UDB V8.2 DB2 Client
- **Note: “or later” is implied when specific versions are mentioned**

Why Do I Want to Get There – Gateway?

- **DB2 uses DRDA as distributed communication protocol**
- **When a DRDA connection is established client and server exchange their version (functional level)**
 - Used functional level is determined by lower of the two (or three)
- **Different versions across a 3 tier environment of client – DB2 Connect Server and DB2 for z/OS can actually be more complicated**
 - Client level may allow new functionality but DB2 Connect Server does not
- **Vendors typically certify a client (and server) version and not configuration**
 - DB2 Connect Server version likely not part of certification statement

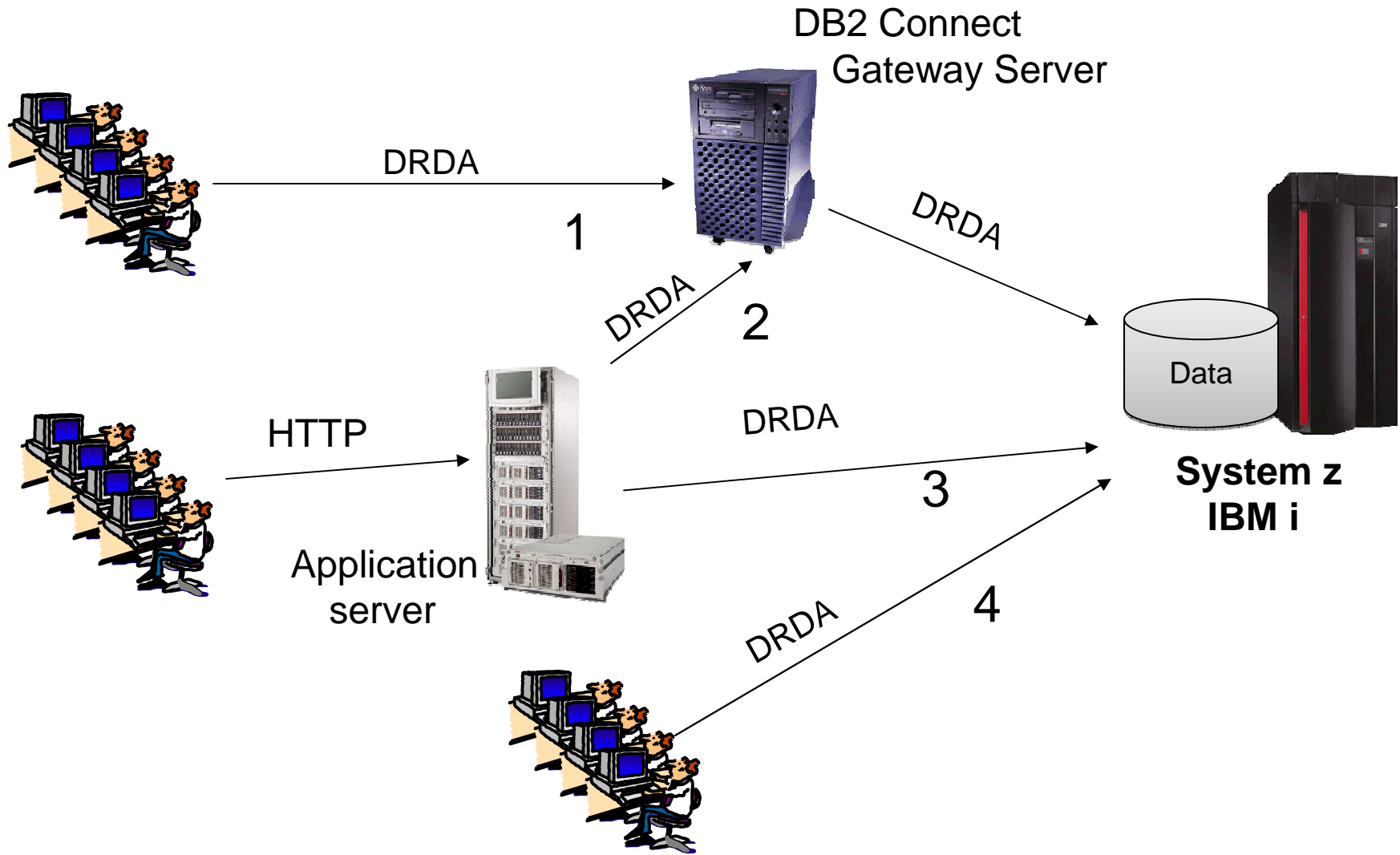
Why Do I Want to Get There – Gateway? (cont.)

- **Simplification of distributed connectivity to DB2 for z/OS**
 - Enhancements in Data Server Clients and DB2 for z/OS
- **Reduced resource consumption**
 - No resources for DB2 Connect Server machines (hardware, software)
 - No installation, maintenance, migration
- **Improved availability as point of failure is removed**
- **Improved response time of distributed applications**
 - 15-30% elapsed time improvement measured in labs
 - OLTP will see the biggest gains

Why Do I Want to Get There – License Keys?

- In a DB2 Connect Server configuration, DB2 Connect entitlement is stored in the DB2 Connect Server and not individual clients
- When migrating to direct client connectivity, DB2 Connect entitlement needs to be stored at each client
- Need to establish plans to rollout DB2 Connect entitlement together with client rollout
- Access entitlement check at the DB2 for z/OS server supported in DB2 Connect 9.7 FP3
 - DB2 Connect Unlimited Edition for z/OS only
 - Applicable to all APIs

DB2 Connect Configurations



Why I No Longer Need a DB2 Connect Server

- **Historically used for protocol conversion between DB2 client (DB2RA) and DB2 z/OS server (DRDA)**
 - Removed with DB2 Connect V8
- **Sysplex Workload Balancing available in drivers**
 - Java available since V8
 - Non-Java available since DB2 9.5 FP3
 - Supports seamless reroute for higher availability
- **Connection Control available on DB2 10 for z/OS**
 - Historically application connections can be mapped to fewer DB2 connections
 - DB2 supports over 100,000 inactive connections with no DB2 impact
 - DB2 provides mechanisms to control max connections, threads and timeout on application server level through the use of server profiles
 - Abstraction layer for application connections
 - DB2 supports improved connection aliases allowing you to dynamically control which members are used for balancing without requiring any DB2 restart

Why Would I use DB2 Connect Server

■ Two Phase Commit

- Many transaction managers still require a DB2 Connect Server for two phase commit if they use a dual transport model
 - For example, Tuxedo, Encina
- DB2 and MS DTC are two transaction managers that do not require a DB2 Connect Server

■ Federation

- Homogeneous Federation is possible with DB2 Connect Server
 - Use of nicknames to other DB2 and Informix data servers

Mechanics of Getting There

- **Migration from client connecting via a DB2 Connect Server to directly connecting to DB2 for z/OS can be done gradually**
 - One client at a time
- **Starting point is generally application servers**
 - Highest concentration of connections for workload balancing
 - Best bang for buck by removing a network hop
- **Still a benefit even if the DB2 Connect Server is co-located on the same machine as the application server**

Mechanics of Getting There – z/OS Server

- **Enable DVIPA and Sysplex Workload Balancing if connecting to a DB2 Data Sharing group**

- **Review DB2 for z/OS distributed configuration and adjust if needed**
 - Zparm
 - WLM policies

Mechanics of Getting There – Client

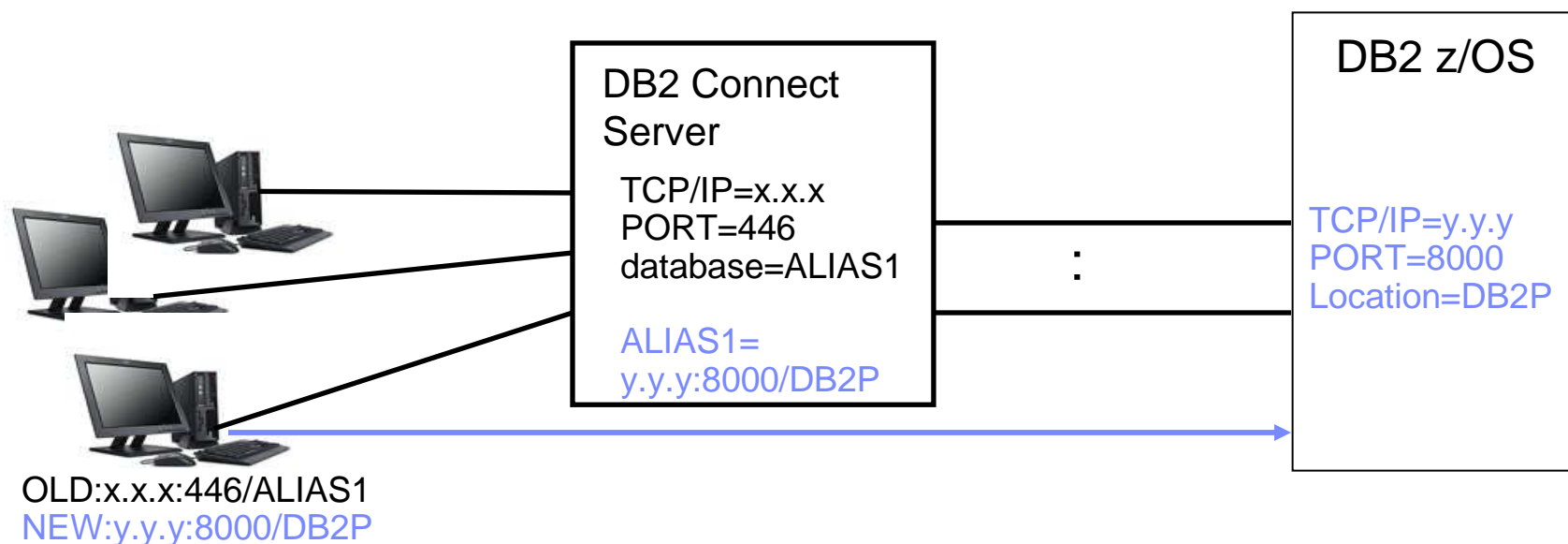
- **Upgrade client machine to latest driver level and smallest driver package**
 - While this step could be optional, it is recommended and can be performed at any time ahead of migration off DB2 Connect Server
- **Ensure client has DB2 Connect license entitlement**
 - Preferred mechanism is Unlimited Edition server based license key
 - Alternative is distributing your DB2 Connect edition license file to each client workstation
- **Change application connectivity to point directly at DB2 for z/OS server**
 - DVIPA plus location name
 - Connection string
 - Data Source
 - `db2dsdriver.cfg` file
 - `db2cli.ini` file
 - Database directory for instance based clients

DB2 Connect Configurations (cont.)

- **DB2 Connect licensing is required in all configurations**
- **DB2 Connect Server does not change licensing, it is only a possible component of all editions except DB2 Connect Personal Edition**

Client Configuration

- Client connectivity information needs to change from pointing to the DB2 Connect Server to pointing to DB2 for z/OS
- If DB2 for z/OS is a Data Sharing group, DVIPA plus location should be used
- If certain applications should only access some members in the data sharing group, DVIPA plus location alias needs to be used
- DB2 10 for z/OS will support dynamic start and stop of location alias

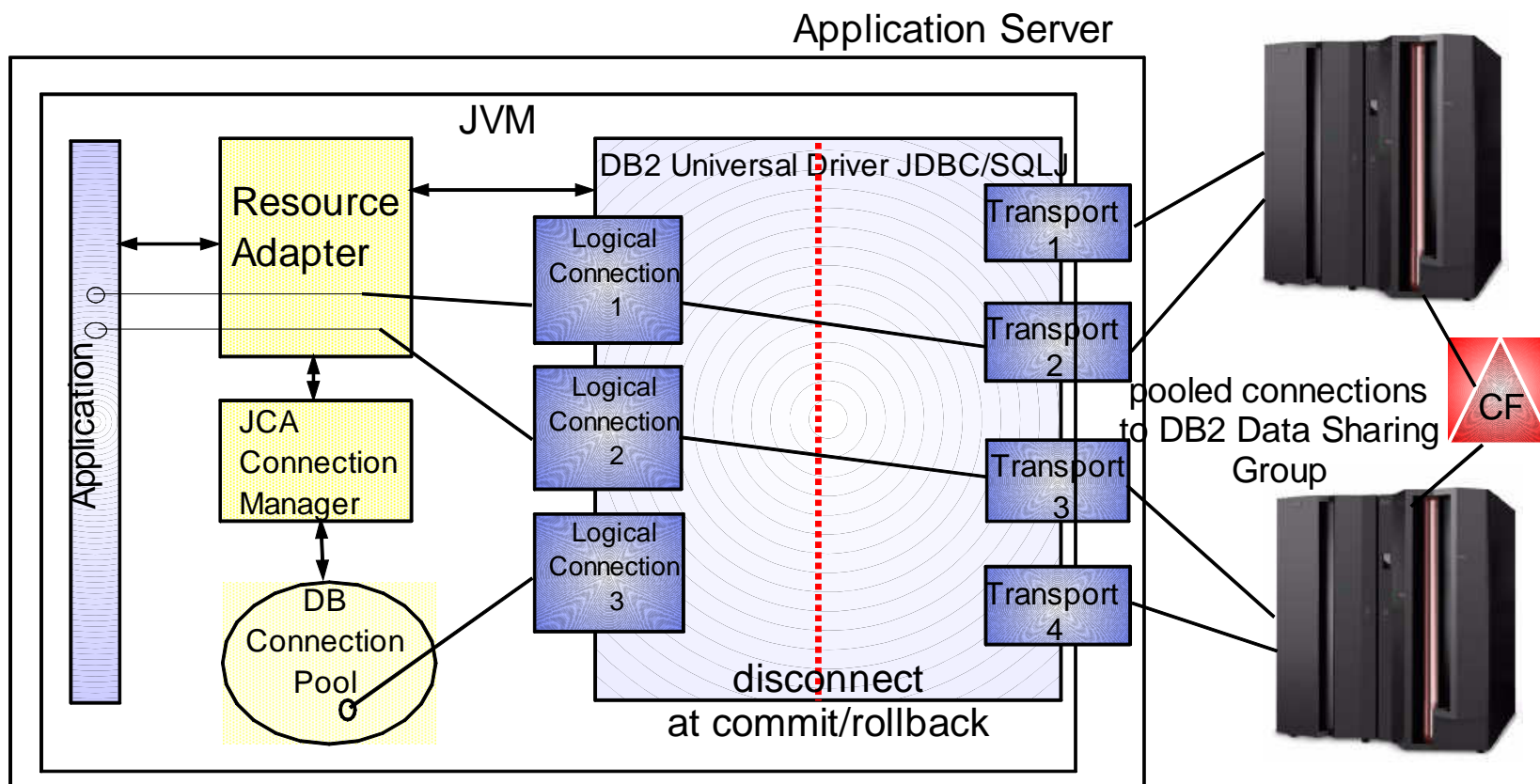


Summary - Where Do I Want to Be?

- **Lightweight client packages utilized throughout my enterprise**
 - IBM Data Server Driver Package (ds driver)
- **Direct connection to my DB2 for z/OS or IBM i server**
 - No DB2 Connect Server to maintain
- **Server based DB2 Connect license key**
 - DB2 Connect Unlimited Edition for z/OS only

Sysplex Workload Balancing

- Transaction level workload balancing supported in
 - JDBC/SQLJ since V8
 - ODBC, CLI and .NET since V9.5 FP3
 - DB2 Connect Server since V7



Fault Tolerant Sysplex Support

- **Initial connection to DB2 data sharing group URL**
 - Recommended to be a distributed DVIPA - this virtual IP address resolves to any available member based on WLM
 - It always works if at least 1 member is up
- **The initial connection (and periodic replies) returns a member IP list with WLM weights**
- **Any following connection uses client sysplex workload balancing algorithm to determine which member to use if reuse is OK**
 - No open WITH HOLD cursor
 - No declared global temporary tables must exist
 - Used declared global temp tables must be explicitly or implicitly dropped
 - No reference to packages bound with KEEP DYNAMIC YES
 - Rollback always leaves a connection in a reusable state

DB2 Connect Function Rollout

- ODBC / CLI and .NET Features

Overview

- **New function rollout limited to Fixpack stream of current release only**
 - V9.5 FP3 was last set of enhancements for V9.5
 - Current enhancements are in V9.7 stream
- **All active releases fully supported**
 - APARs backfit based on customer request and severity
- **Unless stated, function applies to**
 - DB2 for z/OS V8 and later
 - DB2 for IBM i V5R4 and later

Leading up to V9.1

■ Cross API

- Single client instance install
- DB2 Runtime Client (> 100 MB)
- DB2 Client (almost 500 MB)
 - DB2 Connect Personal Edition

■ ODBC / CLI

- Terms used interchangeably

■ .NET

- Framework (FW) 1.0, 1.1
- Visual Studio (VS) VS.NET 2002, VS.NET 2003
- 32-bit only

V9.1

- **Cross API**
 - Multiple copy support (concept of default copy)
 - XML datatype (DB2 for z/OS V9 only)

- **.NET**
 - FW 2.0, VS 2005
 - Dropped FW 1.0
 - DB2ResultSet (scrollable, server locking semantics)

V9.1 FP2

- **.NET**
 - 64-bit Provider
 - End to End SQL Procedure debugging for z/OS V9 servers
 - Single step from application into SQL stored procedure
 - Merged call stacks

V9.5

- **ODBC / CLI**

- cli driver package available
 - Approx 15 MB, designed for ISVs to embed

- **.NET**

- VS 2008 supported
- FW 3.0, 3.5 supported

V9.5 FP1

- **Cross API**
 - 18 character location names
 - Previously only 8 characters supported

V9.5 FP2

- **Cross API**
 - SSL

- **ODBC / CLI**
 - Interleaved SQLPutData for dealing with stream data

- **.NET**
 - Enterprise Library (via CodePlex)
 - Entity Framework public beta
 - VS 2008 WPF and WWF integration
 - End to end SQL procedure debugging for z/OS V8 servers

V9.5 FP3

- **Cross API**
 - IBM Data Server Driver Package (ds driver)
 - db2dsdriver.cfg configuration file introduced

- **ODBC / CLI**
 - SQLCreatePackage API (similar to bind command)

- **.NET**
 - pureQuery static profiling (DB2 Connect Advanced Edition)
 - Entity Framework at FW 3.5 SP1 level
 - Schema node in VS Server Explorer

V9.5 FP3

■ Cross API

- z/OS Sysplex exploitation
 - Client reroute and workload balancing
 - Functions previously only available with DB2 Connect Server
 - Seamless failover
 - Failures in first SQL can be seamlessly rerouted to alternate server without application awareness
 - Configurable in all client packages with dsdriver.cfg file
 - Primary scenario is application servers
 - Performance benefit by removing gateway

V9.7

- **ODBC / CLI**
 - Interleaved fetch and rollback during streaming

- **.NET**
 - Remove any accidental db2cli.ini support
 - Dropped FW 1.1
 - Program Name and Program ID client information properties
 - VS tooling for pureQuery static profiling

V9.7 FP1

■ **Cross API**

- Environment variable for config file location
- Ability to reload config file for ACR section
- Failback to primary support for client affinity

■ **ODBC / CLI**

- pureQuery static profiling
- Interleaved insert with streaming

■ **.NET**

- Full alter support for procedures in VS
- VS 2008 LINQ over XML reference (z/OS only)

V9.7 FP2

- **ODBC / CLI**
 - Command line to register ODBC data source

- **.NET**
 - 32-bit and 64-bit coexistence
 - Entity Framework Filtering
 - Override for with hold cursors

V9.7 FP3a

■ Cross API

- Unlimited Edition server based license key (z/OS only)
- DB2 for z/OS V10 exploitation
 - Binary XML
 - Timestamp precision plus timezone
 - Currently committed semantics
 - Extended indicators
 - Explain modes

V9.7 FP3a (continued)

- **Cross API**
 - Performance Expert Extended Insight integration

- **ODBC / CLI**
 - Network statistics API
 - Retrieve last member used on connection
 - Instance based client support with dsdriver.cfg

- **.NET**
 - Entity Framework canonical functions
 - .NET 4.0, VS 2010, Entity Framework 4.0 beta

V9.7 FP4

- **Cross API**
 - Sysplex migration support for z/OS V10

- **ODBC / CLI**
 - db2cli32 in 64-bit builds
 - db2cli to report on unrecognized settings
 - CLI load enabled for async operation

- **.NET**
 - .NET 4.0, VS 2010, Entity Framework 4.0 GA
 - High precision timestamp and decfloat mapping into dataset
 - Instance based client support with dsdriver.cfg

db2dsdriver.cfg File

- **Introduced for ds driver in V9.5 FP3**
- **Added to instance based client packages:**
 - V9.7 FP3a for CLI
 - V9.7 FP4 for .NET
- **Stored in the “application data”/cfg directory by default**
 - Environment variable available to set location
- **Unrecognized keywords silently ignored**
 - Allows one file across multiple client package versions
 - Syntax errors and invalid values do cause errors

db2dsdriver.cfg File

- **Contains:**
 - Location of a database – name, ip and port
 - Mapping of one or more aliases (dsn) to a database
 - Settings for a dsn, database, or globally
 - For example isolation level, current schema

- **Also contains settings for WLB and ACR**
 - Applicable for all client packages from V9.5 FP3
 - Can only be set in the database section

- **db2cli tool will validate against schema**
 - db2cli validate –dsn samplea [-connect]

db2dsdriver.cfg File

■ Example:

```
<configuration>
  <dsncollection>
    <dsn alias="samplea1" name="sample" host="gross.torolab.ibm.com" port="50000">
      <parameter name="executionMode" value="static"/>
      <parameter name="allowDynamicSQL" value="false"/>
      <parameter name="sqlLiteralSubstitution" value="enable"/>
      <parameter name="pureQueryXML" value="testclicop.pdqxml"/>
    </dsn>
    <dsn alias="samplea" name="sample" host="gross.torolab.ibm.com" port="50000">
    </dsn>
  </dsncollection>
  <databases>
    <database name="sample" host="gross.torolab.ibm.com" port="50000">
      <parameter name="CommProtocol" value="TCPIP"/>
    </database>
    <database name="samples" host="gross.torolab.ibm.com" port="50001">
      <parameter name="CurrentSchema" value="OWNER1"/>
      <wlb>
        <parameter name="enableWLB" value="true"/>
        <parameter name="maxTransports" value="50"/>
      </wlb>
      <acr>
        <parameter name="enableACR" value="true"/>
      </acr>
    </database>
  </databases>
  <parameters>
    <parameter name="GlobalParam" value="Value"/>
  </parameters>
</configuration>
```

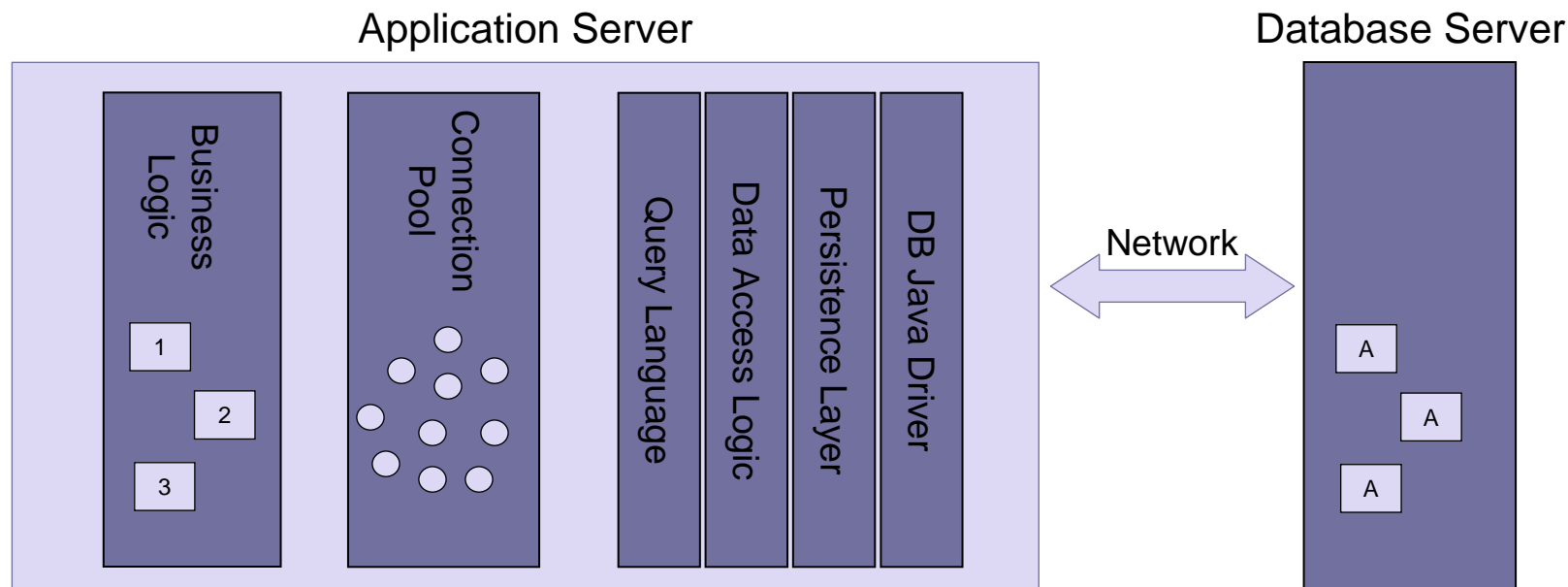
Useful Links

- **db2cli validate page**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp?topic=/com.ibm.swg.im.dbclient.install.doc/doc/r0058814.html>

- **DB2 Connect Unlimited Edition server based license key**
 - <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/topic/com.ibm.db2.luw.licensing.doc/doc/t0057375.html>

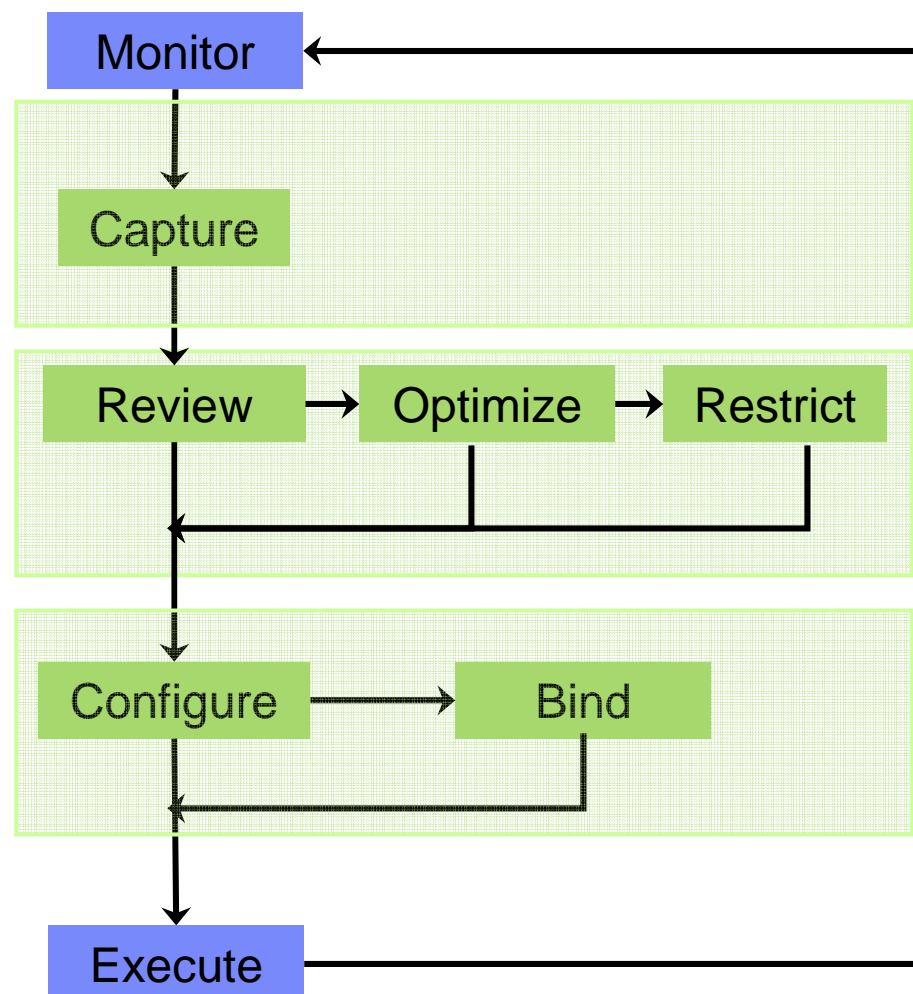
Layered Application Stack Challenges

- **Simplify development, but ...**
 - Challenge problem resolution
 - Impact performance
 - Obscure impact analysis
 - Impede capacity planning
- **Layers obscure linkages**
 - Where is the problem?
 - Which application is running?
 - Which application does this SQL belong to?



Query management: Proactively closing the loop

- **Monitor**
 - Review performance
 - Analyze hotspots
- **Capture**
 - Capture performance and application metadata
- **Optimize**
 - Review and share SQL
 - Visualize hotspots
 - Analyze impacts from schema changes
 - Trace SQL to originating source
 - Optimize and replace SQL
 - Create approved SQL list
- **Deploy**
 - Configure execution properties
 - Optionally bind for static execution



Introducing pureQuery

A high-performance, data access platform to simplify developing, managing, securing, and optimizing data access for new and existing applications.

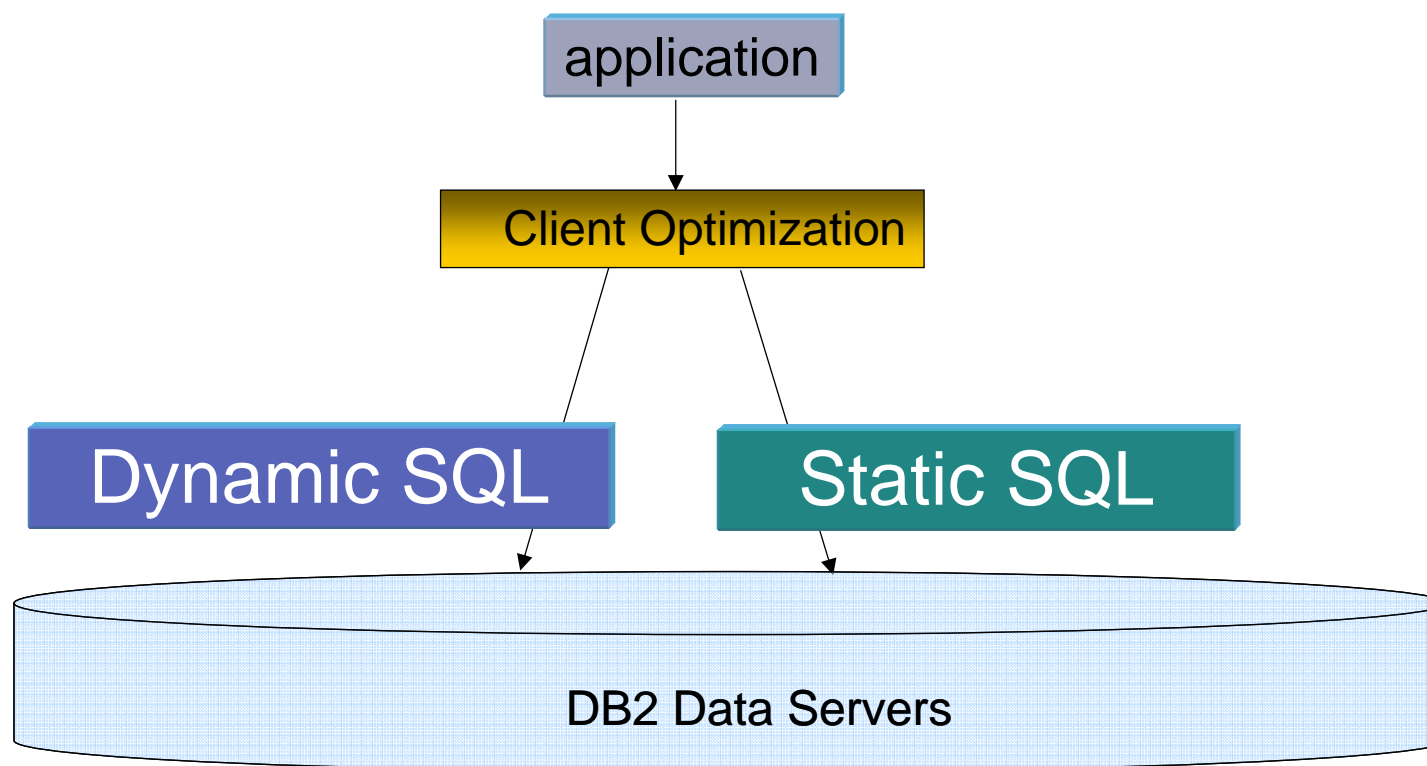


pureQuery Components:

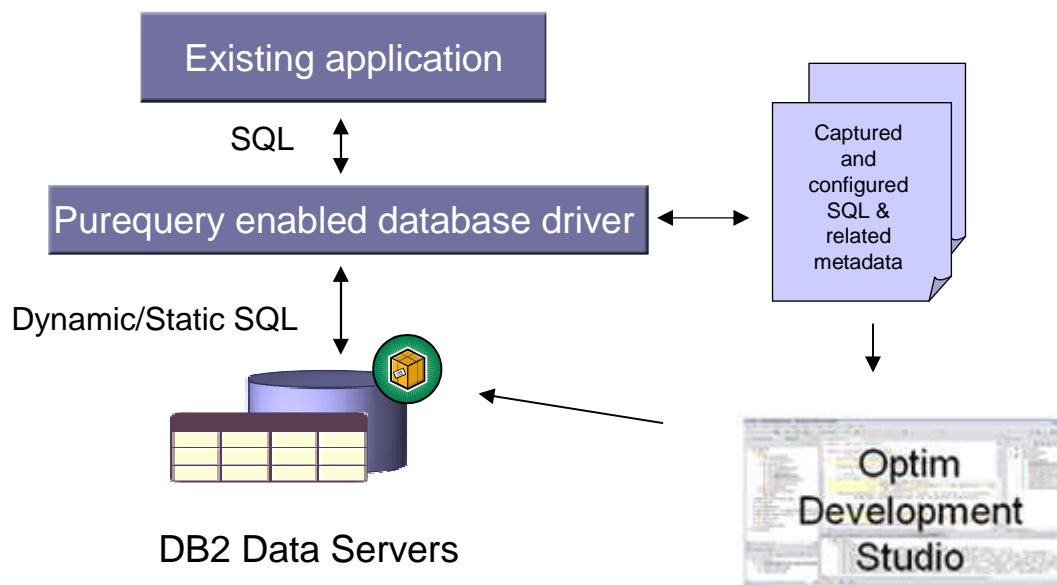
- **Optim Development Studio (ODS)**
 - Integrated development environment with Java and SQL support
 - Improve problem isolation and impact analysis
- **Optim pureQuery Runtime**
 - Flexible static SQL deployment for DB2

pureQuery Client Optimization

- pureQuery client optimization enables dynamic SQL execution and static execution for *existing* applications without any code changes



How does pureQuery client optimization work?



Query management with pureQuery

- **Improve performance**
 - Average of 25% CPU reduction reported by customers
 - Eliminate SQL literal overhead
 - Reduce network round trips for distributed applications

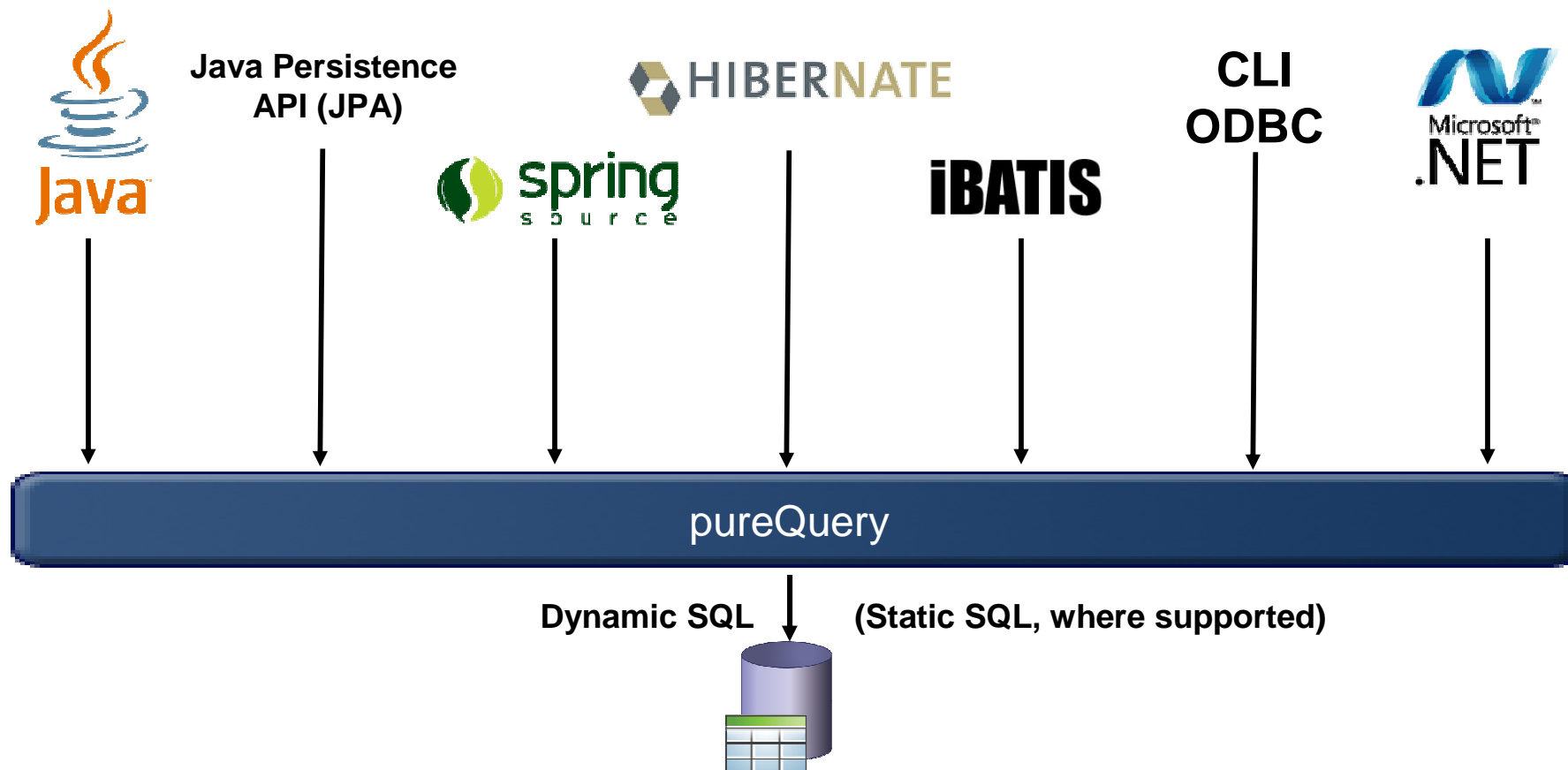
- **Problem determination**
 - Find cause in database and application – even down to line of code
 - Everyone works with a common view
 - Track even Hibernate's generated SQL – “Developers can't hide”

- **High performance SQL – can be reviewed by the DBA**
 - DBA can easily review SQL from developers
 - Reliable performance with locked in access paths
 - Replace problem SQL as needed
 - Java data access managed more like COBOL
 - Static SQL instead of dynamic SQL
 - Control which SQL is run by the app – pre-approved white lists
 - Manage by SQL package

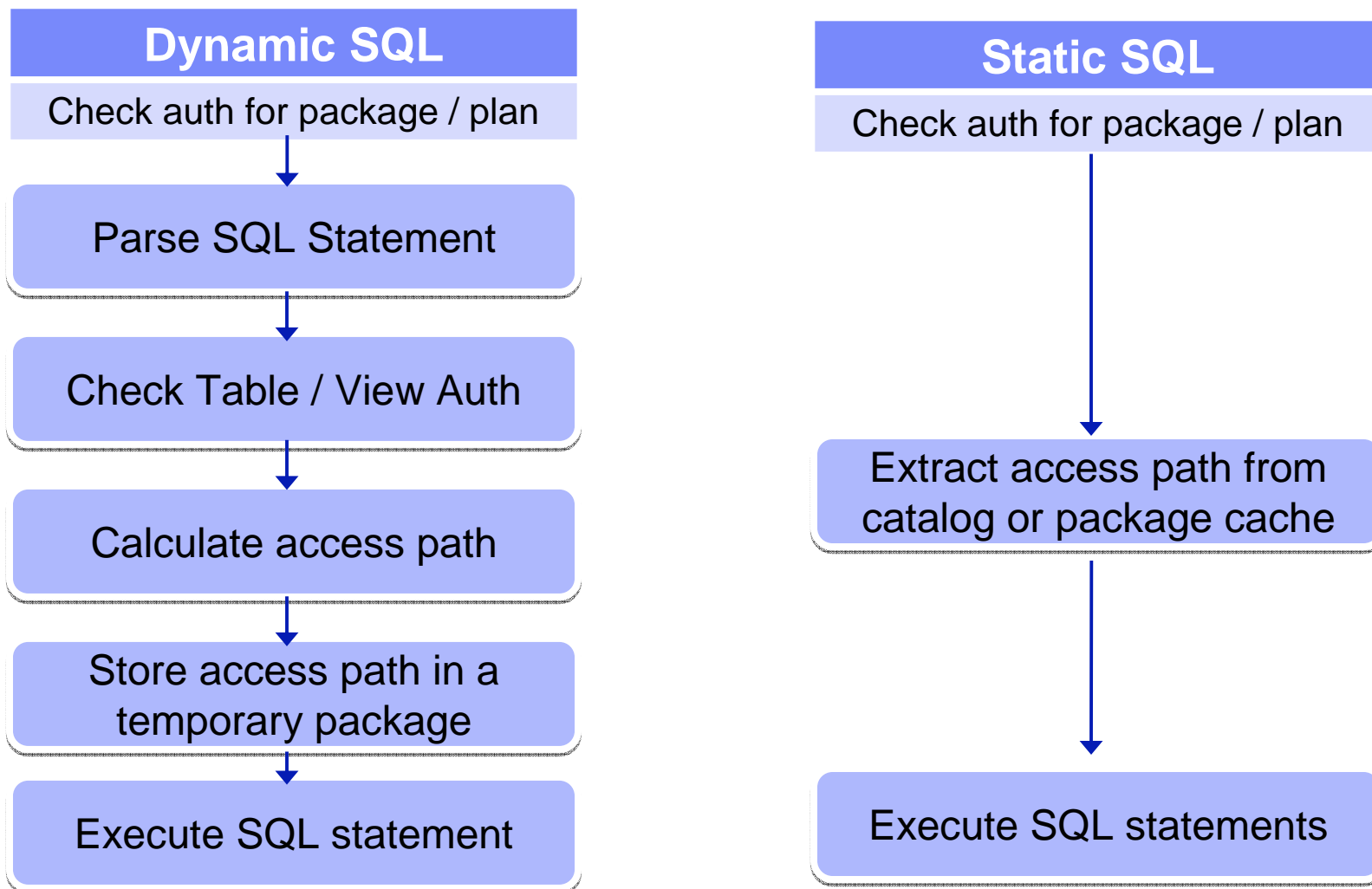
- **Secure access**
 - Grant permissions by SQL package – not database object
 - Eliminate SQL injection risk

Database Access and pureQuery

Many On-ramps for New and Existing Applications



SQL Execution - Dynamic vs. Static

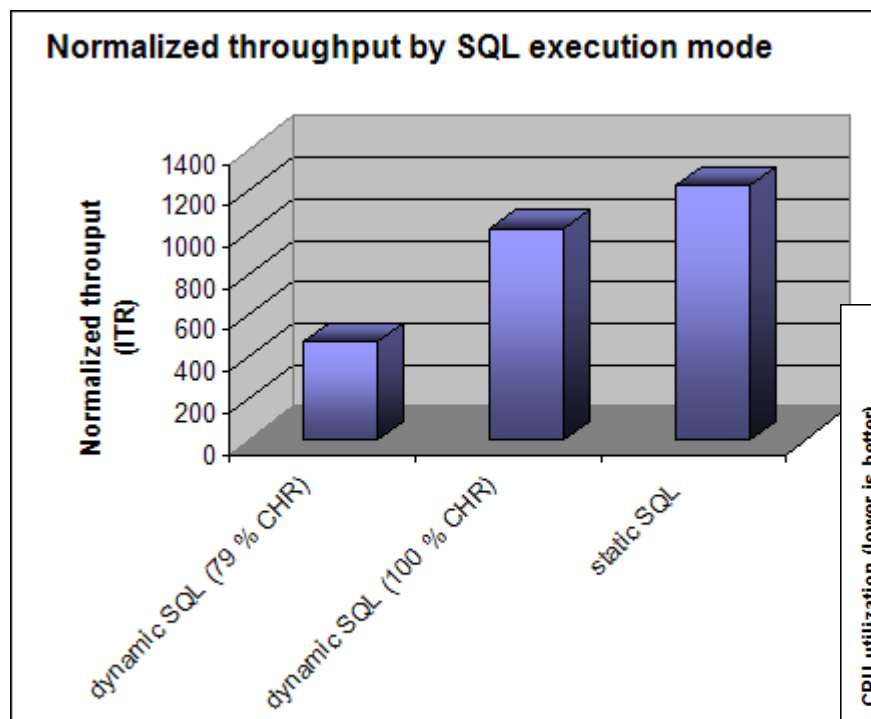


SQL Execution Advantages

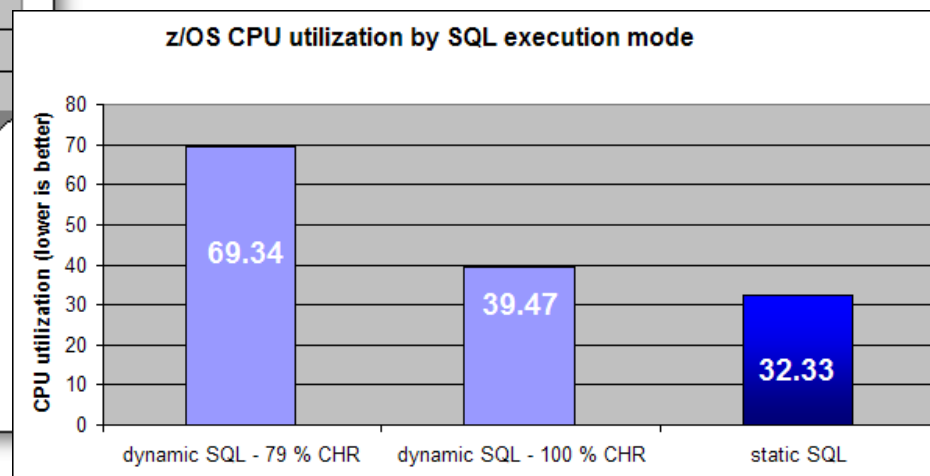
Dynamic vs. Static

Feature	Dynamic SQL (pureQuery, JDBC)	Static SQL (pureQuery, SQLJ)
Performance	Can approach static SQL performance with help from dynamic SQL cache. Cache misses are costly	All SQL parsing, catalog access, done at BIND time. Fully optimized during execution.
Access path reliability	Unpredictable – Any prepare can get a new access path as statistics or host variables change	Guaranteed – locked in at BIND time All SQL available ahead of time for analysis by EXPLAIN.
Authorization	Privileges handled at object level. All users or groups must have direct table privileges – Security exposure, and administrative burden	Privileges are package based. Only administrator needs table access. Users/Groups have execute authority. Prevent non-authorized SQL execution.
Monitoring, Problem determination	Database View is of the JDBC or CLI package – No easy distinction of where any SQL statement came from.	Package View of applications makes it simple to track back to the SQL statement location in the application
Capacity planning, Forecasting	Difficult to summarize performance data at program level.	Package Level Accounting gives program view of workload to aid accurate forecasting.
Tracking dependent objects	No record of which objects are referenced by a compiled SQL statement	Object dependencies registered in database catalog

How well does it work?



- IRWW – OLTP application
- Application runs in IIS and accesses DB2 for z/OS



- Throughput during static execution increased by 159% over dynamic SQL execution assuming a 79% statement cache hit ratio

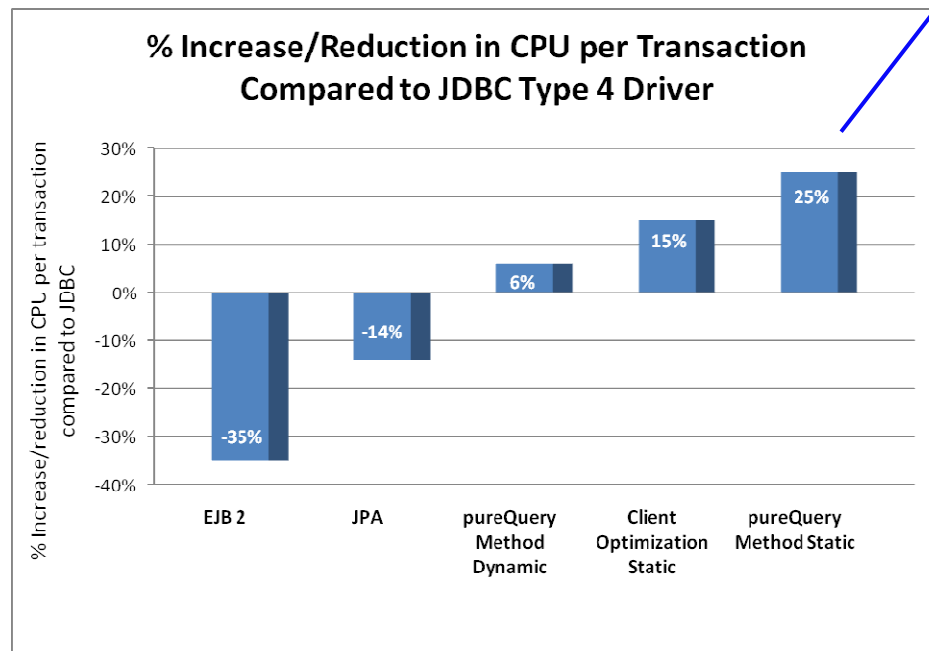
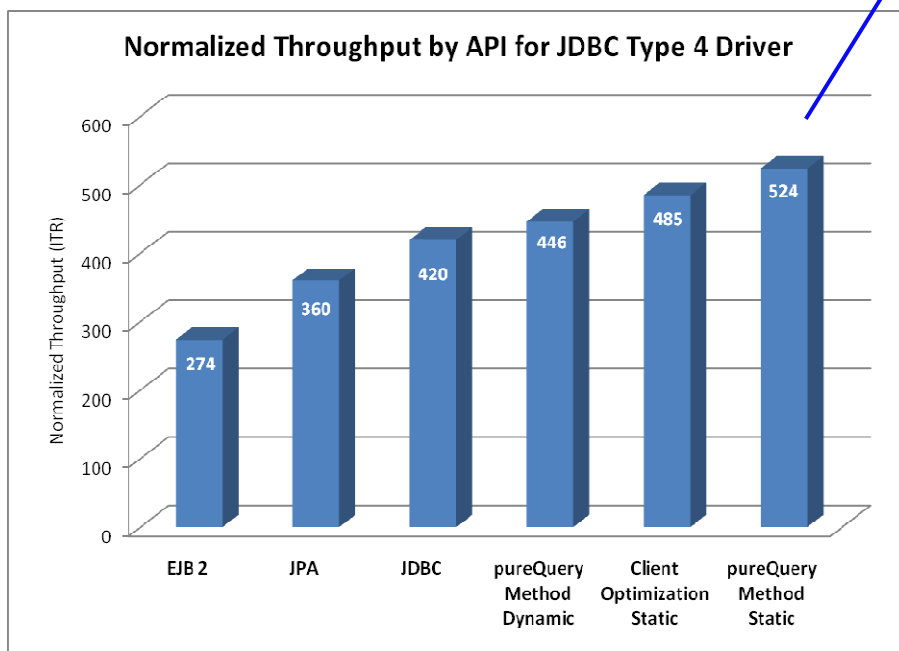
**Any performance data contained in this document were determined in various controlled laboratory environments and are for reference purposes only. Customers should not adapt these performance numbers to their own environments as system performance standards. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.*

Improving Throughput with pureQuery, a z/OS Example

- In-house testing shows 15% - 25% reduction in CPU costs over dynamic JDBC
 - Read about [pureQuery Runtime for z/OS Performance](#)
 - IRWW – an OLTP workload, cache hit ratio between 70 and 85%

Preset application path never varies at runtime

Reduced CPU per transaction means more growth on existing resources



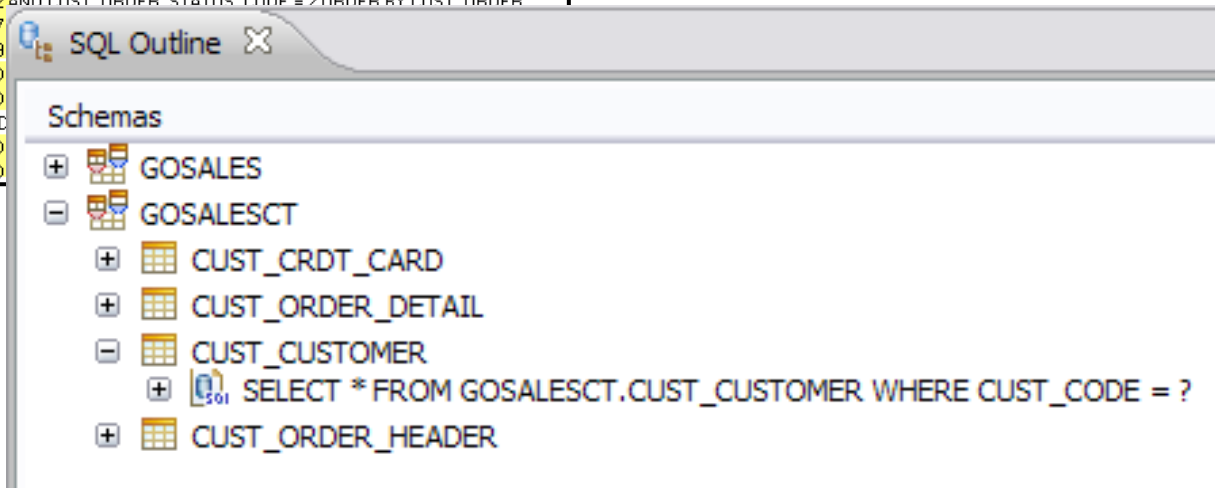
Improve Performance Without Changing Code- e.g. literal replacement

- Review SQL coming from applications

```

SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 35213 AND ORDER_METHOD_CODE = 35213
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 31842
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 27368 AND CUST_CC_ID = 27368 ORDER BY CUST_ORDER_DATE ASC
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 20022 AND CUST_ORDER_STATUS_CODE = 2 ORDER BY CUST_ORDER...
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 48679
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 22302 AND CUST_ORDER_STATUS_CODE = 2 ORDER BY CUST_ORDER...
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 11161 AND CUST_ORDER_STATUS_CODE = 2 ORDER BY CUST_ORDER...
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 18494 AND ORDER_METHOD_CODE = 18494
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 40932 AND CUST_CC_ID = 40932 ORDER BY CUST_ORDER_DATE ASC
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 31968 AND CUST_ORDER_STATUS_CODE = 3 ORDER BY CUST_ORDER...
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7652 AND CUST_ORDER_STATUS_CODE = 2 ORDER BY CUST_ORDER...
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 4427
SELECT * FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149
SELECT COH.CUST_ORDER_NUMBER, COH.CUST_ORDER_DATE, COH.CUST_ORDER_STATUS_CODE FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149
SELECT COUNT(*) FROM GOSALESC.T.CUST_ORDER_HEADER WHERE CUST_CODE = 7149

```



- Consolidate SQL by replacing literals with parameter markers at runtime
- Optimize database resources (e.g. dynamic statement cache) usage

Gaining more application insights – and utilizing them

The screenshot displays the IBM Optim Developer interface with several callouts highlighting key features:

- Correlate with issuing code:** Points to the Java code in the editor, specifically the SQL execution line: `if(statement.execute("SELECT CUST_CODE, STDDEV(ORD_STOT_COST) AS TOTAL, CUST_CODE FROM GOSALESC`.
- Visualize captured SQL:** Points to the SQL Outline view, which lists various SQL queries.
- Visualize and compare execution metrics:** Points to the Performance Data Set table, which shows execution statistics for different queries.
- Replace SQL without changing the application:** Points to the Data Source Explorer, which shows a list of database tables and views.
- Correlate with data source attributes:** Points to the Data Source Explorer, highlighting the connection between application code and data source attributes.
- Execute, tune, share, trace, export SQL:** Points to the context menu for a SQL query in the SQL Outline, listing actions like 'Run SQL', 'Tune SQL', and 'Export SQL to File...'

Schemas	Number of Times Run	Total Time	Max
GOSALESC			
CUST			
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	1	90.92	90.92
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	2	297.06	273.61
SELECT CUST_CODE, CUST_CITY FROM GOSALESC.CUST ORDE	3	20.09	7.99
SELECT CUST_CODE, CUST_FRST_NAME, CUST_LAST_NAME, CU	1	2.70	2.70
SELECT count(CUST_CODE) FROM GOSALESC.CUST	14	61.04	40.82
SELECT count(CUST_CODE) FROM GOSALESC.CUST where C	14	61.04	40.82
ReportGenerator			
CUST_CODE			
CUST_ORD			
SELECT CUST_CODE, STDDEV(4	7.11	2.18
SELECT ORD_NBR AS ORDER	5853	3335.63	46.98
SELECT CUST_CODE, SUM(ORD	3	5.80	2.38
SELECT CUST_CODE, AVG(ORD	4	5.17	1.40
SELECT CUST_CODE, CORRELA	3	3.24	1.20