

Db2 12 and Continuous Delivery What's New from the Optimizer?

Terry Purcell *IBM Silicon Valley Lab* Session code: A04 Tues June 5th 2018

Db2 for z/OS

#IDUGDb2





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





Db2 12 High-level Query Performance Focus

- Query focus based upon newer workloads (modern applications)
 - Complex views or table UDFs
 - UNION ALL
 - Outer joins
 - Join predicates with (stage 2) expressions
 - CASE expressions, CAST functions, scalar functions
- General Bottlenecks
 - Sort/work file reductions
 - Reducing prepare cost and frequency
 - List Prefetch
 - I/O performance
 - Reduce unnecessary prefetch scheduling
- NOTE: 100% zIIP offload for parallel child tasks





4

Significant CPU Savings for Db2 Query Workloads







What are the Workload Characteristics?







6

What are the Workload Characteristics?



5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90% 0%

15% - 25% reduction in ET and CPU

- Traditional query applicationsSort intensive and workfile processing

 - Lots of aggregating
 Many new Access Paths vs. Db2 11





7

What are the Workload Characteristics?



0% 5% 10% 15% 20% 25% 30% 35% 40% 45% 50% 55% 60% 65% 70% 75% 80% 85% 90%

• 0%-15% reduction in ET and CPU

- Simple query workloads
- SQL where vast majority of CPU and ET are due to data scanning (IDAA candidates)
 As % drops *Fewer new Access Paths vs. Db2 11*



Db2 12 for z/OS with IDAA



• Db2 12 CPU savings target

- Operational (in transaction) analytics
- (complex) OLTP

IDAA focus

- Ad-hoc queries
- Complex queries scanning large volume of data
- ETL acceleration/virtual transformation

#IDUGDb2



What was New in 2017?

- Continuous Delivery adds new challenges
 - Common to all companies wanting to modernize their existing applications

2017 focus within Db2 development

- Build delivery & test infrastructure to support continuous delivery (without compromising quality)
- Improve customer Db2 deployment/provisioning
- Next gen z HW enhancements
- Db2z Optimizer
 - Less "fixed in next release" optimizer enhancements more adept to delivery in service (Continuous delivery)
 - Small number of enhancements listed in this presentation (but you have to look!)
 - Strategic initiative(s) to enhance overall access path reliability/stability in choosing best performing plan (for 2018)



#IDUGDb2





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





UNION ALL and Outer Joins

- UA or OJ Performance Challenges
 - Can arise if intermediate results are materialized (written to a workfile) "without" filtering predicates being available before materialization.

• Db2 12 high level solutions

- Reorder OJ tables to avoid materializations
- Push join predicates inside UA legs or OJ query blocks
 - 2017 adds multi-level predicate pushdown
- Sort of outer to ensure sequential access to inner
- Bypass workfile usage when materialization required
- Push ORDER BY and FETCH FIRST into UA legs
- Trim unnecessary columns and tables





Pushdown (+more) Eg: V12 Outer Join with Transparent Archive

Original Query:

SELECT T1.C1, T2.C2 FROM T1 LEFT JOIN T2 ON T1.C1=T2.C1;







Example: V12 ORDER BY / FFnR with UNION ALL

View:

CREATE VIEW V1 AS (SELECT * FROM T1 UNION ALL SELECT * FROM T2 UNION ALL SELECT * FROM T3);

Original query:

SELECT C1 FROM V1 ORDER BY C1 FETCH FIRST 10 ROWS ONLY;

ORDER BY and FFNR Pushed down to base tables

Db2 V12 rewrite:

SELECT C1 FROM (SELECT C1 FROM T1 ORDER BY C1 FETCH FIRST 10 ROWS ONLY UNION ALL SELECT C1 FROM T2 ORDER BY C1 FETCH FIRST 10 ROWS ONLY **UNION ALL** SELECT C1 FROM T3 ORDER BY C1 FETCH FIRST 10 ROWS ONLY) AS V1 ORDER BY C1 FETCH FIRST 10 ROWS ONLY;

Merge of rows from each leg Fetch first 10





Extend LEFT JOIN Pruning

- Db2 10 delivered LEFT OUTER JOIN table pruning
 - If right table guaranteed unique (due to unique index or DISTINCT/GROUP BY) and no columns required for final result

SELECT T1.* FROM T1 LEFT JOIN T2 ON T1.C1 = T2.UNIQUE_COL;

SELECT DISTINCT T1.* FROM T1 LEFT JOIN T2 ON T1.C1 = T2.C1;

• Db2 12

- Extends table pruning to
 - Views, table expressions
 - Where no columns required and no duplicates returned
- Improves pruning of unreferenced columns from materialized results





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





Generic Query Challenge

- Generic "search screen" challenge for query optimizers
 - Filtering could change each execution
 - Making it impossible to choose the 1 best access path

```
• SELECT *
FROM CUSTOMER
WHERE LASTNAME LIKE ?
AND FIRSTNAME LIKE ?
AND ADDRESS LIKE ?
AND CITY LIKE ?
AND ZIPCODE BETWEEN ? AND ?
```

- Fields NOT searched by user will populate with the whole range:
 - LIKE '%' or BETWEEN 00000 AND 99999
- Fields searched will use their actual values
 - LIKE 'SMITH' or BETWEEN 95141 AND 95141





Runtime Adaptive Index

- Allow list prefetch plans to quickly determine index filtering @ execution time
 - Adjust (adapt) @ execution time based on determined filtering
 - Without requiring REOPT(ALWAYS)
 - · For list prefetch or multi-index OR-ing
 - Earlier opportunity to fallback to tablespace scan if large % of table to be read
 - For multi-index AND-ing
 - Reorder index legs from most to least filtering
 - Early-out for non-filtering legs, and fallback to rscan if no filtering
 - Quick evaluation done based upon literals used (e.g. LIKE '%')
 - Further (more costly) evaluation of filtering deferred until after 1 RID block retrieved from all participating indexes
 - Provides better optimization opportunity while minimizing overhead for short running queries





Runtime Adaptive Index

- Not limited to the search screen challenge
 - Any query where there exists high uncertainty in the optimizer's estimates
 - Range predicates
 - JSON, Spatial & Index on expression

```
SELECT * FROM TAB1 WHERE COL1 < ? AND COL2 < ? AND COL3 < ?
INDEXES: IX1(col1), IX2(col2), IX3(col3)</pre>
```

- Performance
 - List PF up to 23% CPU reduction (when failover to rscan needed)
 - Multi-index OR up to 20% CPU reduction (when failover to rscan needed)
 - Multi-index AND up to 99% CPU reduction for re-ordering to put most filtering leg 1st





UDF Caching

- Improve performance of UDFs by caching results by
 - maintaining a hash table that stores UDF result
 - hash key is the UDF input values
 - Scope
 - UDFs that are DETERMINISTIC and NO EXTERNAL ACTION
 - Avoid invoking UDF and input expression again for the same input
 - Caching is within the scope of an individual UDF instance only (cache is not shared)
 - Features
 - all datatypes (except XML,LOB,ARRAY) and expressions with up to 1 column
 - adaptively turn off caching if no benefit seen (check every 1024 executions)
 - Performance
 - Could reduce UDF time up to 99% for low cardinality inputs





Sort Minimization for Partial Order with FFnR

- Problem
 - With ORDER BY & FETCH FIRST n ROWS ONLY (FFnR)
 - Only avoid ORDER BY sort if an index is chosen that supports the ORDER BY
 - Allowing only the FFnR number of rows to be fetched
 - If sort cannot be avoided
 - Full result read/sorted

• Db2 12

- Can utilize an index that provides order on leading columns of the ORDER BY
 - Reducing the number of rows fetched/processed if sort cannot be avoided





Partial Order with FFnR - Example

• Example

SELECT * FROM T1 ORDER BY C1, C2 FETCH FIRST **10** ROWS ONLY

INDEX1 (C1)

Index entries: 1,1,1,2,2,2,3,3,3,4,4,4,**4**,**5**.....999999

- Pre Db2 12
 - Fetch all rows and sort into C1, C2 sequence

10th row Stop fetching

- Db2 12
 - Once 10th row is reached, fetch until next change in C1
 - 13 rows read, 12 rows fetched/sorted in this example





Sort and Sparse Index – Reducing Resource Consumption

- Sort avoidance for OLAP functions with PARTITION BY clause
 - ROW_NUMBER, RANK, DENSE_RANK
- More sorts in-memory (continuing theme since V9)
 - Sort tree scales with zparm SRTPOOL
- Sparse index support for vargraphic
- Memory reductions
 - Sort
 - Remove constants from the GROUP BY/DISTINCT sort key
 - Reduce sort key redundancy
 - Sparse index
 - Avoid duplicating key information when key=data and fixed length key
 - Trim trailing blanks for VARCHAR/VARGRAPHIC
 - Trim prefix if all keys have the same prefix





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





Targeted Stage 2 Predicate Improvements

Sparse index support for stage 2 join predicates

T1.col2 = SUBSTR(T2.col2, 1, 10)

- Excludes CASE expressions, DECFLOAT, different CCSIDs, FULL OUTER JOIN
- Indexability for VARBIN/BINARY predicates when operand lengths don't match
 - Includes indexability for Index On Expression built on VARBIN/BINARY datatypes

CREATE INDEX EMPLOYEE_NAME_SORT_KEY ON EMPLOYEE (COLLATION_KEY(LASTNAME, 'UCA410_LDE', 600));

Indexability for correlated subqueries on row permissions for Insert/Update

CREATE PERMISSION RP1 ON LINEITEM T1 FOR ROWS WHERE EXISTS (SELECT 1 FROM ORDER T2 WHERE T1.ORDERKEY=T2.ORDERKEY) ENFORCED FOR ALL ACCESS ENABLE





Merge for Table UDFs similar to Views

- Merge rather than materialization for Table UDFs
 - Defined as DETERMINISTIC and NO EXTERNAL ACTION
 - Similar to performance of views
- Support indexability of predicates passed in as TUDF parameters

```
CREATE FUNCTION TUDF(in_date1 CHAR(10), in_date2 CHAR(10))
...
SELECT O_ORDERPRIORITY, COUNT(*)
FROM ORDERS
WHERE O_ORDERDATE >= DATE(in_date1) <<< Was STAGE 2. Now indexable
AND O_ORDERDATE < DATE(in_date2) <<< Was STAGE 2. Now indexable</pre>
```

```
SELECT * FROM table(TUDF('2017-07-01', '2017-10-01')) as TAB1;
```





Extend NPGTHRSH to default stats

- NPGTHRSH zparm prioritizes matching index access over tablespace scan
 - If NPAGESF is < NPGTHRSH</p>
 - NPGTHRSH recommendation is generally 10 or less
- Pre Db2 12
 - Default stats (-1) = 501 pages. Thus default stats may result in tablespace scan chosen.
- Db2 12
 - Compare -1 to NPGTHRSH rather than 501
 - Applies when
 - table and indexes both have default stats
 - table has stats but one or more indexes have default stats





List Prefetch and Hybrid Join

- Optimizer enhancements to list prefetch due to improved runtime
 - Adaptive index has better runtime estimation of actual predicate filtering
 - Allowing earlier fallback to tablespace scan when poor filtering
 - RID sort performance improvements
- Improved Hybrid Join (HBJ) as choice within parallelism
 - Enables HBJ (without sort new) for all clusterratios based upon cost
- Significant performance improvement possible for unclustered data





Improve Filter Factor for CURRENT DATE/TIMESTAMP

- Problem
 - Predicates including special registers CURRENT DATE/TIMESTAMP are treated similar to host variables/parameter markers
 - Result is a default Filter Factor estimate
- Db2 12

• COL op CURRENT DATE / TIMESTAMP (incl non-col expression)

• Will resolve the expression to actual value for filter factor purposes.

```
SELECT * FROM T1
WHERE SALES_DT > CURRENT_DATE - 7 DAYS
AND SALES DT < CURRENT DATE;</pre>
```





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





Dynamic Plan Stability

- Provide "static SQL" like performance stability for repeating cached dynamic SQL
 - Store dynamic SQL and cache structures in the Db2 catalog
 - Load cache from catalog on cache miss, catalog hit
- Allow more stable, predictable query performance across...
 - Exit / re-entry to statement cache
 - System recycle
 - RUNSTATS
 - Db2 maintenance
 - Release migration
 - Across members of data sharing group
 - Heavy "cache-load" periods can see significant performance improvement full prepares replaced with loading cache from disk.





Static Plan Stability Enhancements

- Free inactive package copies while package is allocated and in use
 - Allow FREE of INACTIVE only
- Allow selective FREE of ORIGINAL or PREVIOUS

APREUSE source

Supports APREUSE of PREVIOUS or ORIGINAL in 1 step





Statement Concentration BIND Parameter

• [RE]BIND PACKAGE

>>-[RE]BIND PACKAGE----->

• CONCENTRATESTMT:

• NO (default)

• Do not enable statement concentration. Literals in dynamic SQL are not replaced.

YES

• Enable statement concentration. Literals in dynamic SQL are replaced by '&'.





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12







RUNSTATS Enhancements for Improved SQL Performance

- Clusterratio formula improvements
- Db2 12 default is that RUNSTATS will NOT invalidate cache
 - Exception UPDATE NONE REPORT NO
- Profile support for inline stats
- Automated COUNT for FREQVAL
 - Allow Db2 to collect until values no longer skewed (rather than count 10) see next slide
- Optimizer to automatically update statistics PROFILE
 - Specify USE PROFILE to automatically pick up optimizer recommendations



RUNSTATS Automated Count

- When collecting FREQVAL (frequencies)
 - What is the best "count" to use?
 - COLGROUP(C1) COUNT 10? COUNT 20?
 - The answer is:
 - Keep collecting until the data is "no longer" skewed
- Db2 12 adds this capability
 - Exclude the COUNT n keywords and Db2 will automate the COUNT (with a max of 100)
 - NOTE: (APAR PI94111) Optimizer stats externalization utilizes this feature







Optimizer Externalization of Missing Stats







RUNSTATS FREQVAL Performance



- Optimizer FREQVAL recommendations will (likely) increase RUNSTATS cost
 - Given that the optimizer has recommended collection of this statistic the hope is that they increased RUNSTATS cost is returned by improved SQL performance
- FREQVAL collection
 - Leading index columns cost = cheap
 - Non-indexed or non-leading index columns cost <> cheap

V11 APAR PI74408, V12 APAR PI76730 Improve performance and zIIP offload for RUNSTATS with non-indexed FREQVAL

V11 4-way Z13 zOS 2.1 Table of 100 milions rows, 20 partitions, 6	V1	1	V11+	DECK	delta v11 vs deck		% z	IIP eligible
indexes	СРИ	elapsed	CPU	elapsed	CPU	elapsed	V11	V11+deck
LOAD STATISTICS TABLE COLGROUP	1092.34	614.03	667.88	437.96	-39%	-29%	39%	64%
REORG STATISTICS TABLE COLGROUP	1046.22	556.81	568.85	352.98	-46%	-37%	39%	59%
RUNSTATS COLGROUP	705.93	437.86	254.99	256.52	-64%	-41%	45%	100%





- Db2 12 Performance Focus
- UNION ALL & Outer Join Enhancements
- Runtime Optimizations (incl sort/WF, sparse index)
- Predicate and Cost Model
- Plan Stability
- RUNSTATS & Optimizer driven profile updates
- Getting the most from Db2 12





Plan Management Pre-Migration Prep





• First – there is no reason NOT to use PLANMGMT=EXTENDED

- But There is NO capability to FREE only an ORIGINAL copy PRIOR to Db2 12
 - FREE PACKAGE PLANMGMTSCOPE(PLANMGMTINACTIVE) frees original and prev
- The ORIGINAL can become stale
 - The idea is to keep a "good and stable" (recent) backup in case of emergency
- Once you are comfortable with the CURRENT copy
 - Consider FREE PACKAGE PLANMGMTSCOPE(PLANMGMTINACTIVE)
 - Immediately before migration to 12 (or before migration to any release!)
 - So that 1st REBIND in new release will save the prior release's CURRENT copy to be the ORIGINAL





Give the Db2 12 Optimizer a Chance.....eventually

• **REBIND** with APREUSE is recommended immediately after migration

Reduces risk of access path regression

• Rebind *without* APREUSE to see full potential for CPU savings

- 15+ internal workloads measured
 - Queries with new AP's consistently were the biggest winners
- Typical performance gains for *new AP's*
 - 10%-99% reduction in CPU and ET
- Typical performance gains with **NO AP change**
 - 0%-20% reduction in CPU and ET





Migration Considerations – Old Statistics



• Old (stale) statistics

- Customers often run "specialized" stats as a one-off to try to solve an issue or as a prior default.
 - These old statistics can become stale and cause access path issues
 - Simplest way to find these is to look for tables with different STATSTIMEs in SYSCOLDIST
- Db2 11 delivers
 - RUNSTATS reset option
 - Sets relevant catalog values to -1, and clears tables such as SYSCOLDIST

RUNSTATS TABLESPACE db-name.ts-name TABLE table-name RESET ACCESSPATH

- Recommend running "regular" RUNSTATS after RESET
- Blog post
 - http://www.worldofdb2.com/profiles/blogs/ensure-you-have-a-simple-base-of-statistics-before-exploiting-the





- Culprits are usually multicolumn statistics or histograms
 - Quick check using this SQL
 - Remove ASAP since PROFILE usage can incorrectly assume these are critical to collect
 - Cleanup becomes critical in Db2 12
 - Or if using SET PROFILE FROM EXISTING STATS (V10-12)

```
SELECT TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
, MIN(STATSTIME), COUNT(*)
FROM SYSIBM.SYSCOLDIST CD
WHERE STATSTIME < CURRENT TIMESTAMP - 1 MONTH
AND
   (TYPE IN ('C', 'H') OR NUMCOLUMNS > 1
  OR STATSTIME < CURRENT TIMESTAMP - 1
YEAR)
AND NOT EXISTS
(SELECT 1
FROM SYSIBM.SYSINDEXES I
WHERE I.TBCREATOR = CD.TBOWNER
AND
      I.TBNAME = CD.TBNAME
AND
      CD.STATSTIME BETWEEN I.STATSTIME - 8 DAYS
                       AND I.STATSTIME + 8 DAYS)
AND NOT EXISTS
(SELECT 1
FROM SYSIBM.SYSTABLES T
WHERE T.CREATOR = CD.TBOWNER
AND
      T.NAME = CD.TBNAME
AND
      CD.STATSTIME BETWEEN T.STATSTIME - 8 DAYS
                       AND T.STATSTIME + 8 DAYS)
GROUP BY TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
ORDER BY TYPE, NUMCOLUMNS, TBOWNER, TBNAME, NAME
WITH UR;
```





Optimizer Config Recommendations (due to Db2 12)

- ZPARMS
 - NPGTHRSH
 - Consider setting to 1 to preference index access for tables/partitions with CARDF=0 or -1
 - MAXRBLK
 - Db2 10 increased the default to 400000. Many customers still below this value.
 - PLANMGMT
 - Db2 9 set default to EXTENDED. Many customers do not use.
- Client Config
 - Disable default of CURSOR HOLD used by ODBC/JDBC
 - As WITH HOLD disables DB2 in-memory sorts
 - ODBC
 - Can be changed by setting CURSORHOLD=0 in db2cli.ini file
 - JDBC
 - Can be changed by setting resultSetHoldability=2





Db2 for z/OS News from the Lab blog http://ibm.biz/db2znews



Get the latest news from the IBMers who design and build Db2!

- New capabilities in Db2 12 for z/OS continuous delivery
- Enhancements in Db2 11 for z/OS
- Helpful tips and best practices from Db2 for z/OS development
- Join the conversation
 - Subscribe to follow the blog
 - Become a member to comment
 - Follow us on Twitter: <u>@Db2zLabNews</u>





Copyright © 2018 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. 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. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

- Performance data contained herein was generally obtained in a controlled, isolated environments. 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.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and Disclaimers





- Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
- The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.
- IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System[™], FASP®, FileNet®, Global Business Services ®, Global Technology Services ®, IBM ExperienceOne[™], IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics[™], PureApplication®, pureCluster[™], PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, 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.

Notices and Disclaimers Continued



Terry Purcell IBM tpurcel@us.ibm.com

Session code: A04

Please fill out your session evaluation before leaving!

#IDUGDb2