

DB2 10 for z/OS

Cut Cost & Improve Performance

SOMMARIO

1. SEZIONE I – INTRODUZIONE AL DB2 10 FOR Z/OS	1-5
1.1 SOMMARIO	1-5
2. SEZIONE II – PERFORMANCE AVAILABILITY	2-7
2.1 MANY PERFORMANCE ENHANCEMENTS	2-7
2.2 NEW OPTIMIZER ACCESS PATH RANGE-LIST INDEX SCAN	2-7
2.3 OPTIMIZER USES MORE PARALLELISM	2-7
2.4 LIST PREFETCH IMPROVES INDEX ACCESS	2-8
2.5 OPTIMIZER NO LONGER CHANGES ACCESS PATH BECAUSE OF RID POOL OVERFLOWS	2-8
2.6 OPTIMIZER DOES MORE DURING STAGE 1 SQL EVALUATION	2-8
2.7 OPTIMIZER DETERMINES WHICH ACCESS IS MORE CERTAIN	2-8
2.8 DYNAMIC STATEMENT CACHE ATTRIBUTES IMPROVEMENTS	2-9
2.9 IMPROVED DDF TRANSACTION FLOW	2-9
2.10 WORKFILE ENHANCEMENTS	2-9
2.11 MEMBER CLUSTER OPTION	2-9
2.12 UNIVERSAL RANGE-PARTITIONED TABLE SPACE	2-9
2.13 BUFFER POOL ENHANCEMENTS	2-10
2.14 NEW AND IMPROVED ON-LINE SCHEMA CHANGES	2-10
2.15 INCLUDE NON-UNIQUE COLUMNS WITHIN A UNIQUE INDEX	2-11
2.16 NEW HASH SPACE AND ACCESS METHOD	2-11
2.17 DB2 CATALOG ENHANCEMENTS	2-11
3. SEZIONE III – SCALABILITY, SIMPLIFICATION, SECURITY	3-12
3.1 FULL 64-BIT RUNTIME ENVIRONMENT EXPLOITED	3-12
3.2 LARGE INCREASE IN THE NUMBER OF USERS UP TO 20,000	3-12
3.3 PARALLEL INSERTS INTO MULTIPLE INDEXES	3-12
3.4 PLAN STABILITY - PACKAGE PRESERVATION	3-12
3.5 RUNSTATS IMPROVEMENTS AND AUTO STATS - REAL TIME STATISTICS STORED PROCEDURES	3-13
3.6 IMPROVED AND FINER GRAIN ACCESS CONTROL - SECURITY AND REGULATORY COMPLIANCE	3-13
3.7 SUPPORT FOR ROW AND COLUMN ACCESS CONTROL	3-13
4. SEZIONE IV - APPLICATION ENABLEMENT	4-14
4.1 PUREXML ENHANCEMENTS	4-14
4.2 SUPPORT FOR XML DATE AND TIME	4-14
4.3 BINARY XML SUPPORT	4-14
4.4 FASTER STREAMING OF XML AND LOBS	4-14
4.5 XML DEFINE (NO) FOR LOB AND XML TS	4-15
4.6 TEMPORAL QUERIES AND THEIR BUSINESS ADVANTAGES	4-15
4.7 TIMESTAMP, TIME ZONE AND OTHER DATA TYPE ENHANCEMENTS	4-16
4.8 SUPPORT FOR TIMESTAMP WITH TIME ZONE	4-17
4.9 ACCESS TO CURRENTLY COMMITTED DATA	4-17
4.10 SQL COMPATIBILITY IMPROVEMENT - EXTENDED INDICATOR VARIABLE	4-17
4.11 EXTENDED SUPPORT FOR IMPLICIT CASTING	4-18
4.12 ENHANCED SCALAR FUNCTION SUPPORT	4-18
4.13 SQL PROCEDURAL LANGUAGE ENHANCEMENTS	4-18
5. SEZIONE V – DATA WAREHOUSING	5-19
5.1 SUPPORT FOR TEMPORAL TABLES AND VERSIONING	5-19
5.2 OLAP FUNCTIONALITY BUILT DIRECTLY INTO DB2	5-19
5.3 ADVANCED BUSINESS ANALYTICS	5-19
6. SEZIONE VI – REDUCED TOTAL COST OF OWNERSHIP (TCO)	6-21
7. APPENDICE	7-22
8. RIFERIMENTI	23
8.1 ARTICOLI	23

8.2	PRESENTAZIONI.....	23
8.3	REDBOOKS	23

1. Sezione I – Introduzione al DB2 10 for z/OS

1.1 Sommario

Il DB2 10 per z/OS è un enorme passo avanti nella tecnologia di database per i suoi miglioramenti nelle prestazioni, scalabilità, disponibilità, sicurezza e integrazione applicativa. La sua tecnologia fornisce un chiaro vantaggio competitivo attraverso i suoi continui miglioramenti in SQL, XML e attraverso l'integrazione di funzionalità di business intelligence.

Il DB2 10 sfrutta i recenti progressi nella tecnologia dei chip, dei dispositivi di archiviazione e la capacità di memoria attraverso lo sfruttamento esteso dell'architettura System z a 64-bit per ridurre la richiesta di CPU, migliorare le prestazioni e ridurre potenzialmente e drasticamente il totale dei costi di gestione.

La riduzione della CPU del DB2 10 è stata accolta con interesse in tutto il mondo spingendo le Aziende a sfruttare qualsiasi fattore e miglioramento per modificare la propria linea di riferimento/misurazione.

I test iniziali in DB2 10 hanno mostrato miglioramenti del tempo di esecuzione riducendo il consumo della CPU fino al 5-20%. Inoltre il DB2 10 può gestire da cinque a dieci volte più utenti simultanei, fino ad arrivare a 20.000, fornendo rilevanti miglioramenti nella capacità, scalabilità e migliori prestazioni complessive per ogni tipo di applicazione.

Capacità, scalabilità e prestazioni continuano a essere i punti di forza di DB2 per z/OS attraverso il pieno sfruttamento dell'architettura System z a 64 bit, i miglioramenti DB2, come una minor durata dell'elaborazione sfruttando la tecnologia dei dischi a stato solido, **in memory workfile**, **index insert parallelism improvements** e **better SQL/XML access-paths**, forniscono una notevole riduzione dei costi e della CPU e miglioramenti delle prestazioni senza dover apportare modifiche alle applicazioni.

Il DB2 è continuamente disponibile poiché è aumentata la possibilità di modificare parametri on-line, più utilities possono girare contemporaneamente ed è più facile la gestione dei processi. Il DB2 10 è sempre disponibile, anche quando le modifiche al database sono in corso, le transazioni continuano a essere elaborate. Queste nuove funzionalità di modifica al database combinate con le nuove possibilità per l'esecuzione delle utilities in parallelo consentono di rendere la finestra di elaborazione e manutenzione sempre più piccola.

Gli aspetti della sicurezza, la conformità alle normative e i miglioramenti delle capacità di controllo sono inclusi nel DB2 10. E' aumentata la sicurezza del DB2 10 poiché è esteso il modello basato sui ruoli introdotto nel DB2 9. Il DB2 10 consente una maggiore granularità per le autorizzazioni al fine di consentire l'accesso ai dati separato dal tipo di domanda, database e sistema. Il DB2 10 fornisce la flessibilità di amministrazione per specifiche impostazioni del ruolo di sicurezza al fine di prevenire l'esposizione di accesso ai dati da applicazioni non autorizzate o da altri amministratori. Tale modello di sicurezza basato sul ruolo combinato tra LBAC (label-based) e RCAC (controllo di accesso di colonna e mascheramento) oppure dalla crittografia delle informazioni più sensibili rendono l'intero ambiente di database più sicuro per il business aziendale. Tutte queste caratteristiche offrono controlli più severi e permettono una maggiore flessibilità di sicurezza fornendo enormi possibilità di auditing.

Migliora l'integrazione e la portabilità di applicazioni SQL e XML. Questo miglioramento consente di avvicinare le applicazioni DB2 con altre applicazioni di altri vendor. Nuovi miglioramenti riguardano la gestione dei timestamp con i fusi orari e la precisione dei timestamp fino al pico-secondo. Quest'aspetto è importante per quelle aziende che operano in situazioni internazionali (per esempio le banche).

La gestione delle versioni dei dati tra le tabelle correnti e i dati delle tabelle storiche è disponibile anche con il DB2 10. Tale gestione fornisce la flessibilità per migrare i dati operativi correnti in una tabella di dati storici basata su periodi temporali. Ciò consente di migliorare le prestazioni di un'applicazione operativa, mantenendo i dati storici disponibili per eventuale audit.

Anche le capacità di data-warehousing e di business intelligence dei dati sono ora costruite direttamente nel DB2 10 attraverso le nuove funzionalità di tabelle temporali e la capacità di SQL per il calcolo multiimensionale (OLAP) moving sums, averages e aggregates.

Il DB2 10 offre miglioramenti efficaci per quanto riguarda la gestione degli XML, migliorando la flessibilità delle applicazioni, l'usabilità e le prestazioni. Attraverso la capacità di sostituire, cancellare o inserire i nodi del documento XML, di gestire multiple versioni degli XML schema documents e di utilizzare il

formato XML binario pre-token, il DB2 10 fornisce una serie di importanti prestazioni di XML e miglioramenti di gestione.

Nuovi miglioramenti riguardano le utilities: supporto dei data types "date" e "time", supporto di parametro XML all'interno di funzioni e procedure SQL consentono una maggiore flessibilità di applicazione per utilizzare XML all'interno qualsiasi soluzione di architettura applicativa.

Il DB2 10 per z/OS offre prestazioni, scalabilità, disponibilità e sicurezza migliorando l'integrazione tra le applicazioni. Riducendo i costi e avvalendosi di una tecnologia superiore, il prodotto IBM DB2 10 resta il leader tecnologico e il migliore database da scegliere per oggi e domani.

2. Sezione II – Performance Availability

2.1 Many performance enhancements

Per gli utenti di database relazionali, le prestazioni del database sono fondamentali. Il DB2 10 per z/OS sfrutta l'architettura a 64bit per ridurre il consumo di CPU fino al 5-10% immediatamente out-of-the-box, senza modificare le applicazioni. La CPU può essere ulteriormente ridotta fino al 20% una volta che tutti i miglioramenti previsti nel DB2 10 sono utilizzati nel modo New Function Mode. Ecco perché IBM e il DB2 10 continuano a essere leader del mercato dei database.

Disponibilità, scalabilità, sicurezza, conformità, l'integrazione delle applicazioni, XML e SQL contengono tutti i miglioramenti delle prestazioni. Tutte queste migliorie forniscono un ambiente operativo migliorato, rendendo più facile l'amministrazione, riducendo il costo totale di proprietà per il business.

La performance è l'enfasi maggiore del DB2 10. Molti miglioramenti consentono la riduzione immediata del CPU per le applicazioni. Con la migrazione alla nuova versione ed eseguendo il deploy e il re-bind delle applicazioni nel nuovo ambiente, le applicazioni possono sfruttare molti dei miglioramenti delle prestazioni senza cambiare l'applicazione. Il DB2 10 sfrutta completamente l'architettura a 64 bit, nuovi access-path, l'ottimizzazione di altri access-path e fornisce miglioramenti delle prestazioni right-out-of-the-box.

Il REBIND delle applicazioni è ancora più facile. Le preoccupazioni circa la regressione sono facilmente affrontate con il Plan Stability che introdotto con il DB2 9 è stato ulteriormente migliorato con il DB2 10. Nel caso di malfunzionamenti, con il DB2 10 è possibile associare nuovamente i packages senza rischiare aspetti prestazionali. Il Plan Stability consente agli amministratori di esaminare le prestazioni delle applicazioni, scegliere le migliori e quindi di fissarle all'interno del sistema. Le novità del Plan Stability nel DB2 10 forniscono ulteriore flessibilità per la mettere a punto le prestazioni delle applicazioni e ridurre il consumo di CPU senza modificare le applicazioni.

2.2 New Optimizer Access Path Range-List Index Scan

Il DB2 10 offre un nuovo access-path chiamato Range-List Index Scan. Questo access-path migliora l'elaborazione SQL di un indice quando nella clausola WHERE vengono specificati molti predicati che referenziano lo stesso indice. Per esempio, quando l'SQL ha molteplici OR, IN o altri predicati che fanno riferimento allo stesso indice, l'ottimizzatore ora lo riconosce e analizza l'indice solo una volta, invece di più volte per ogni predicato nella clausola WHERE. Questa novità riduce drasticamente il numero di RID da esaminare, di conseguenza migliorano I/O e le prestazioni della CPU.

2.3 Optimizer uses more parallelism

Il DB2 10 migliora anche altri access-path esistenti attraverso il parallelismo. Questi miglioramenti progettati specificamente per eliminare alcune restrizioni DB2 precedenti, aumentano la quantità di lavoro reindirizzando il carico ai processori zIIP, e distribuendo il lavoro in modo più uniforme di tutti i task paralleli. Tutti questi miglioramenti forniscono nuovi motivi per abilitare il parallelismo.

Il parallelismo migliora le prestazioni delle applicazioni e il DB2 10 ora lo può sfruttare appieno con diversi tipi di query SQL, ad esempio: multi-row fetch, full outer join, common table expression (CTE), table expression materializzata, una function table, una CREATE GLOBAL TEMPORARY TABLE (CGTT), oppure un work-file utilizzato in una View materializzata.

Questi nuovi miglioramenti del parallelismo CP sono attivi quando l'SQL Explain contiene "C" nella colonna parallel mode. I nuovi miglioramenti del parallelismo possono anche essere attivi durante altre specifiche situazioni SQL.

Queste situazioni sono:

- quando l'ottimizzatore decide di scorrere con modalità inversa un indice di tabella

- quando una SQL subquery è trasformata in una join
- quando il DB2 sceglie di fare una **multi-column-ibrid-join** associata a un sort
- quando la tabella principale è ordinata e la join tra la tabella principale e la tabella secondaria è un multi-column-ibrid-join

Altre ottimizzazioni del DB2 10 e miglioramenti di access-path possono aiutare molti aspetti prestazionali delle applicazioni. In DB2 10, l'index lookaside e la sequential detection contribuiscono a migliorare il riferimento di una parent keys, in strutture con integrità referenziale (RI) durante una elaborazione di INSERT. Questo metodo è più efficiente rispetto al controllo di RI dependent data e riduce il consumo di CPU complessivo durante l'attività di INSERT.

2.4 List prefetch improves index access

Nel DB2 10 è utilizzata maggiormente la list prefetch per accedere alle pagine index leaf e non-leaf. Nelle precedenti versioni di DB2, in cui l'indice disorganizzato presentava grandi spazi vuoti tra le pagine non-leaf, accedendo alle voci dell'indice con modalità sequenziale, la lettura di pagine non-leaf degradava molto per effetto di un maggior numero di I/Os sincrone. Il DB2 10 migliora tale meccanismo utilizzando le informazioni contenute nella pagina non-leaf ed eseguendo una list prefetch delle pagine leaf. In tal modo viene eliminata la maggior parte di I/O sincrone e la I/O waits dovuto associato ai grandi spazi tra le pagine non-leaf durante la lettura sequenziale. Questa elaborazione di list prefetch aiuta soprattutto l'esecuzione di query che dipendono da pagine non-leaf e migliora tutti i programmi di utilità verso gli indici come la REORG INDEX, la CHECK INDEX e la RUNSTATS INDEX.

2.5 Optimizer no longer changes access path because of RID Pool overflows

Il DB2 10 migliora anche la gestione delle istruzioni SQL che fanno riferimento a un grande quantità di dati attraverso la list processing. La list processing utilizza un gran numero di record ids (RIDs) e talvolta superano la soglia di memoria prevista per il RID Pool. Nelle versioni precedenti di DB2, il superamento della soglia di memoria del RID Pool causava da parte del DB2 il cambiamento dell'access-path con un tablespace scan. Ora, quando le risorse del RID Pool sono esaurite, la RID list è scritta in un workfile e l'elaborazione continua. Questo miglioramento consente di evitare il tablespace scan, riduce l'impatto di locking e il sovraccarico delle prestazioni.

2.6 Optimizer does more during Stage 1 SQL evaluation

L'ottimizzatore del DB2 10 ora può valutare l'access-path dei predicati SQL con funzioni scalari e non-matching index durante la prima fase (Stage 1). L'ottimizzatore può applicare questi criteri prima della fase 2, ovvero all'inizio del processo di ottimizzazione eliminando il numero di dati che si qualificano (pagine e righe). In questo modo migliora notevolmente il tempo impiegato dalla query e le prestazioni complessive della query.

2.7 Optimizer determines which access is more certain

Nelle precedenti versioni di DB2, l'ottimizzatore valutava per i predicati contenuti nella clausola WHERE, gli indici disponibili per la tabella, inoltre si avvaleva di varie informazioni statistiche per determinare il più efficiente access-path. Con i miglioramenti del DB2 10, l'ottimizzatore fa un'analisi supplementare per determinare l'access-path, valutando i predicati SQL con range, la non uniformità della distribuzione dei dati, l'utilizzo di parameter markers, i literals oppure le **variabili host** dove i valori delle **variabili** sono sconosciuti.

Al momento di scegliere tra due diversi indici, la valutazione di queste variabili fattore di filtro sconosciuti potrebbe rendere la stima dei costi tra due diversi index access-path molto vicina. Analizzando quale delle diverse variabili fattore filtro sono note, permette al DB2 di determinare quale index access-path ha un maggior grado di certezza di successo. Il DB2 potrebbe scegliere un index access-path con un costo leggermente superiore ma che fornirà delle prestazioni al

runtime più sicure. Ciò è particolarmente importante se si considera che le applicazioni possono essere scritte con linguaggi di programmazione differenti e utilizzo di variabili differenti come: i parameter markers, i literals e le variabili host.

2.8 Dynamic Statement Cache ATTRIBUTES improvements

Uno dei più importanti miglioramenti del sistema DB2 10 è la capacità di combinare alcune variazioni di SQL all'interno della **dynamic-statement-cache**. Utilizzando la nuova clausola ATTRIBUTES all'interno dell'istruzione SQL PREPARE, DB2 è ora in grado di riconoscere se l'SQL è lo stesso, tranne che per i valori letterali della clausola WHERE. Questo aiuta il DB2 a capire che queste istruzioni sono già presenti all'interno della cache e quindi di riutilizzare le risorse della cache in precedenza generate per l'istruzione SQL. In tal modo si evita una nuova attività sul catalogo DB2, come la verifica di un oggetto e dell'access-path per un'altra istruzione SQL. Questo aiuta anche a liberare spazio nella cache per altre istruzioni SQL migliorando le prestazioni e il tempo di risposta delle transazioni.

2.9 Improved DDF transaction flow

Le prestazioni delle applicazioni e il traffico di rete delle transazioni è ottimizzato quando le istruzioni SELECT sono codificate utilizzando la clausola FETCH 1 ROW ONLY. Il DB2 ora riconosce questa istruzione SELECT FETCH 1 ROW ONLY e combina le istruzioni SQL OPEN, FETCH e CLOSE in una singola richiesta invece di tre messaggi separati inviati attraverso la rete.

La modifica alla clausola FETCH 1 ROW ONLY migliora anche le prestazioni JDBC e CLI API. Dopo che i dati della query sono recuperati, la clausola FETCH 1 ROW ONLY provoca l'azione predefinita per l'API e chiude le risorse. Il DB2 chiude le risorse indipendentemente dal fatto che un cursore WITH HOLD è stato dichiarato e notifica all'API di cancellare qualsiasi ulteriore FETCH oppure CLOSE richiesta. Questo riduce drasticamente il numero e la quantità di messaggi di rete trasmessi dalle transazioni, migliorando le prestazioni e riducendo al minimo le contese DB2.

2.10 Workfile enhancements

Diversi tipi di istruzioni SQL, come le joins, GROUP BY, e ORDER BY, utilizzano i workfiles per ottenere i risultati. Il DB2 10 ora può valutare semplici query senza workfiles, migliorando le prestazioni complessive e riducendo la durata.

Anche nel database di tipo WORKFILE, il DB2 10 supporta la definizione di tablespace partition-by-growth per supportare tabelle in memoria e migliorare le prestazioni.

Inoltre, il DB2 10 espande la lunghezza record dei file di lavoro fino a 65.529 bytes per consentire la movimentazione di record di risposta di grandi dimensioni.

2.11 MEMBER CLUSTER option

Il DB2 cerca di mantenere il tablespace clustering quando i dati sono inseriti nella tabella. Negli ambienti di data-sharing con un'attività consistente di INSERT, mantenere il tablespace clustering può causare problematiche di locking tra i tablespace nei diversi data-sharing members. Tale contesa provoca un incremento dei cicli di CPU con conseguenti situazioni di stallo e aumento dei tempi di risposta.

Nel DB2 10, i tablespaces **partition-by-range** e **partition-by-growth** hanno un nuovo parametro, MEMBER CLUSTER che consente al DB2 di inserire dati nel primo tablespace disponibile e di ignorare il clustering. In questo modo la contesa sui dati è alleviata, ma va monitorato il tablespace clustering del tablespace. Un degrado del tablespace clustering può avere un impatto negativo sugli access-path e talvolta causare un incremento del random I/O per applicazioni critiche ad alte prestazioni.

2.12 Universal range-partitioned table space

Il DB2 10 continua inoltre a migliorare il tablespace universale range-partitioned. Questo tablespace è la versione aggiornata del classico tablespace range-partitioned con la specifica del segment-size, le impostazioni universali e la capacità. Il tablespace universal range-partitioned è l'obiettivo della migrazione per il classico tablespace range-partitioned che è deprecato e sarà supportato solo per poche altre versioni DB2.

I DBA sono incoraggiati a utilizzare il nuovo universale range-partitioned tablespace invece del classico range-partitioned per sfruttare tutte le nuove funzionalità delle utilities, la disponibilità e i miglioramenti delle prestazioni.

2.13 Buffer Pool enhancements

System z migliora la gestione di allocazione di memoria e il DB2 10 sfrutta questa capacità attraverso i suoi miglioramenti di gestione dei buffer pool. Nelle versioni precedenti, DB2 allocava le dimensioni definite del buffer pool all'avvio per l'utilizzo da parte di tutte le tabelle e indici associati. In DB2 10 il sistema alloca la memoria per il buffer pool come i dati sono portati nel pool di buffer. Ciò mantiene la dimensione del buffer pool al minimo ed è allocato solo ciò che è utilizzato alle applicazioni.

Inoltre, il DB2 10 riduce il latch contention e consente di definire buffer pool più grandi dagli attuali megabyte a gigabyte di memoria per gli oggetti attivi più critici. Questo consentirà di migliorare sicuramente I/O del sistema e di ridurre la contesa.

Il nuovo System z10 può gestire pagine di buffer pool in memoria fino a 1-megabyte. Poiché la dimensione del buffer pool può diventare molto grande rapidamente; una gestione prudente del page size z10 1-megabyte non mancherà di tenere il sistema di paging al minimo.

2.14 New and improved on-line schema changes

Con il DB2 10 continua ad ampliarsi la gamma di modifiche che possono essere applicate al database on-line, per qualsiasi tabella DB2, indice o elemento di tabella. Prendersi cura di un database diventa ancora più semplice. E' possibile alterare on-line un parametro e poi eseguire il processo di Reorg per far recepire la modifica. Il DB2 10 offre una maggiore flessibilità di gestione senza intaccare la disponibilità delle applicazioni per quasi tutte le modifiche che possono essere fatte sul database.

Sono molteplici le modifiche on-line che possono essere apportate a tablespace, tabelle, indici e colonne. Alcune di queste sono: la modifica della grandezza di un file, la modifica della dimensione di pagina di tabelle/indice, la modifica del segment-size, la possibilità di migrare un vecchio tablespace in un tablespace partition-by-growth (PBG) oppure partition-by-range (PBR) oppure in un tablespace universale.

Inoltre, le migliorie introdotte nel DB2 10 a proposito dell'on-line schema changes, sostituiscono i vecchi processi operativi di DROP / RECREATE tablespace / tabella e i processi di REBUILD index con un solo comando di ALTER e un successivo processo di REORG. Quando si utilizza la REORG on-line, il nuovo processo di ALTER e REORG lascia le tabelle disponibili per le attività dell'applicazione, mentre le nuove modifiche sono applicate nel sistema. In tal modo si eliminano i tempi di inattività e aumenta la disponibilità per quelle applicazioni considerate mission-critical in database di grandi dimensioni (VLDB).

Poiché i database continuano a crescere, il carico transazionale aumenta, avere tempo a disposizione per amministrare un database è diventato sempre più complicato. A tal fine sono stati introdotti alcuni miglioramenti nelle utilities, per ridurre i tempi di inattività durante le normali attività di manutenzione.

Ora anche con il DB2 10, i DBA saranno in grado di creare o ricostruire un indice non univoco per tabelle con i LOB senza creare alcun impatto sulle applicazioni. Un altro aspetto è quello di poter aiutare un'applicazione già installata creando un nuovo indice in modo rapido per migliorare notevolmente l'accesso SQL oppure risolvere un problema di prestazioni.

I nuovi metodi per l'applicazione delle modifiche ALTER e REORG introducono il nuovo stato di restrizione del database AREOR per segnalare che ci sono modifiche pendenti. Il nuovo stato del database AREOR con le tabelle del catalogo DB2, con le utilities e il comando DROP PENDING CHANGES fornisce tutte le funzionalità per la gestione dei processi di ALTER e REORG.

2.15 INCLUDE non-unique columns within a unique index

Un'altra novità introdotta col DB2 10, è la capacità di includere altre colonne in indici univoci. Questa nuova funzione del DB2 10 consente di includere colonne non univoche nella definizione di un indice univoco. Prima di questo miglioramento, più indici erano richiesti per l'indicizzazione, uno per il vincolo di univocità e un altro per le colonne non univoche. In questo modo sono eliminati tutti gli I/O supplementari necessari per accedere alla tabella.

2.16 New Hash space and access method

Il DB2 10 introduce anche un nuovo tipo di access-path chiamato Hash Access. Un nuovo spazio Hash supporta questo nuovo access-path. Il DB2 utilizza un algoritmo interno hash con lo spazio Hash per fare riferimento alla posizione delle righe di interesse. In alcuni casi, questo legame Hash diretto riduce l'accesso dati a un singola I/O, diminuendo drasticamente il carico di lavoro della CPU e velocizzando i tempi di risposta dell'applicazione. Le query che utilizzano i predicati di uguaglianza come il numero cliente oppure il numero prodotto sono buoni candidati per l'accesso Hash. Nuovi indici possono essere creati per supportare accessi alternativi.

La definizione di uno spazio Hash richiede una colonna oppure colonne per l'accesso diretto alla chiave Hash. Ogni tabella con accesso Hash ha uno spazio Hash associato oppure uno spazio Hash partizionato. L'accesso Hash richiede un po' di spazio di archiviazione aggiuntivo, ne consegue una riduzione notevole dei tempi di accesso e del carico di lavoro della CPU.

Un altro vantaggio dell'accesso Hash è che non è necessario mantenere un indice clustering oppure avere dati in sequenza. Questo permette di avere una maggiore efficienza durante l'elaborazione delle INSERT e di evitare contese in una ambiente data sharing per il mantenimento di una clustering sequenze oppure clustering index.

Nel caso di utilizzo dell'accesso Hash, sia il parallelismo sia il tradizionale clustering dei dati, non saranno disponibili. Tuttavia, l'accesso Hash è utile per quei database in cui le chiavi univoche stanno già utilizzando predicati di uguaglianza come sul prodotto o cliente ID, o oggetto ID, documenti XML ids per altri tipi di accesso diretto.

2.17 DB2 Catalog enhancements

Il catalogo e la directory DB2 sono stati ristrutturati attraverso la rimozione di strutture speciali e link. Con questa ristrutturazione, il catalogo DB2 utilizza il nuovo tablespace partizion-by-growth, il meccanismo di reordered row format e una tabella per tablespace, ampliando così il numero di tablespace DB2. Questi nuovi tablespace universali DB2 sono definiti con DSSIZE 64 MAXPART 1, locking a livello di riga e contengono alcune colonne CLOB e BLOB per gestire stringhe molto lunghe. Con questa nuova definizione, anche le tabelle di catalogo possono essere gestite come le tabelle di una applicazione con utilities di REORG e CHECK.

Un altro miglioramento riguarda l'eliminazione del lock UTSERIAL che causava contese nelle precedenti versioni DB2 durante l'esecuzione delle utilities. Tale miglioramento, associato ad una riduzione della log latch contention, ad una nuova opzione di lettura per evitare l'attesa durante gli inserimenti e miglioramenti nel thread latching serialization contribuiscono a ridurre molti tipi di contesa di thread DB2. Tutti questi miglioramenti aiutano a diminuire drasticamente le problematiche legate alla contemporanea esecuzione di DDL, BIND, utilities e l'elaborazione complessiva all'interno del sistema di database di applicazioni e, soprattutto, i processi che fanno riferimento al catalogo DB2.

Un altro miglioramento riguarda la possibilità di aggiungere un nuovo LOG nell'inventario del sistema (BSDS). Il LOG appena aggiunto è immediatamente disponibile senza dover necessariamente compiere operazioni particolari in DB2.

3. Sezione III – Scalability, Simplification, Security

3.1 Full 64-bit runtime environment exploited

Il DB2 10 migliora notevolmente la scalabilità in virtù del maggior sfruttamento dell'architettura a 64-bit dell'ambiente System z. Sfruttando l'ambiente a 64-bit e spostando dall'80% al 90% della memoria DB2 che è ora sotto la barra dei 2GB (working storage, EDMPOOL, ect...) sopra la barra di 2GB, il DB2 10 elimina i principali vincoli di memoria all'interno dell'address space DB2DBM1.

3.2 Large increase in the number of users up to 20,000

Le modifiche alla mappatura di archiviazione nell'address space DBM1 riducono la richiesta di memoria al di sotto della barra del 75% o più. Le limitazioni di memoria virtuale all'interno dell'address space DBM1 non sono più un problema. Infatti, il DB2 10 può elaborare fino a 5/10 volte l'attuale carico di lavoro in thread. Inoltre, ci potrebbe essere l'occasione di consolidare gli attuali carichi di lavoro in un numero minore di membri che condivisione dati DB2 oppure di sottosistemi DB2 che sono soggetti a consumo di memoria ECSA.

3.3 Parallel Inserts into multiple indexes

Il DB2 10 migliora anche le prestazioni delle INSERT utilizzando più parallelismo. Quando una SQL INSERT modifica una tabella con molti indici, il DB2 10 esegue il prefetch di più indici in parallelo. Con il parallel I/O per più indici, i processi non restano più in attesa per l'I/O sincrono, riducendo il tempo complessivo del processo di inserimento. Ciò riduce i tempi di possibile conflitto all'interno del sistema e migliora le prestazioni di tutte le applicazioni.

3.4 Plan Stability - package preservation

Con il Plan Stability, introdotto con la versione 9, viene offerto un metodo migliore per gestire la sperimentazione di un nuovo pacchetto applicativo DB2.

Il Plan Stability dà l'opportunità di poter tornare indietro qualora l'associazione dei nuovi pacchetti all'applicazione non risultassero efficienti. Ciò avviene eseguendo il comando di REBIND SWITCH. Il DB2 10 espande questa funzionalità, fornendo la capacità di gestire molte versioni di packages SQL sia statici che dinamici. Il Plan Stability ha la capacità di consentire la gestione di due o tre copie di vecchi pacchetti SQL statici. Il controllo avviene attraverso il parametro DB2 PLANMGMT e i parametri del comando REBIND.

Attraverso il parametro REBIND SWITCH, gli amministratori possono scegliere qualunque versione dei pacchetti presenti. Questo offre al team di sviluppo dell'applicazione la flessibilità di valutare le prestazioni e ottenere la versione più performante del pacchetto implementato. Il Plan Stability prevede anche la possibilità di confrontare gli access-path, fissare un access-path e restituire un messaggio di errore/avviso nel caso in cui l'access-path sia cambiato, e proteggere i pacchetti critici per evitare che siano sostituiti da altri che prevedono access-path meno performanti. Questa miglioria del DB2 10 elimina la necessità di eseguire un REBIND di un'applicazione e aiuta gli amministratori a proteggere le prestazioni delle applicazioni critiche.

3.5 RUNSTATS improvements and Auto stats - Real time statistics stored procedures

Giacché l'ottimizzazione degli access-path sta migliorando notevolmente le prestazioni, la statistica al momento **up-to-the-moment** è di vitale importanza. Il DB2 10 è dotato di un nuovo set di stored procedure per monitorare e raccogliere le statistiche di tabelle e indici. Queste nuove stored procedure eseguono il monitoraggio delle attuali statistiche, e determinare se le nuove statistiche devono essere raccolti e quindi eseguire autonomamente la raccolta al fine di garantire una buona ottimizzazione degli access-path.

Queste stored procedure aiutano soprattutto gli ambienti volatili e possono contribuire in modo dinamico a migliorare l'ottimizzazione degli access-path per leggere le statistiche di index filtering per i predicati utilizzati in SQL WHERE e prendere le migliori decisioni di access-path. Queste stored procedure si assumono l'onere di amministrare oggetti di grandi dimensioni al fine di garantire sempre le migliori prestazioni applicative.

3.6 Improved and finer grain access control - Security and regulatory compliance

Il DB2 10 include miglioramenti su sicurezza e audit. Aumenta la sicurezza basata sui ruoli DB2 prevista dalla versione 9 con una capacità più raffinata di concessione di autorizzazione e privilegi. Questa granularità aiuta a separare la parte di amministrazione e accesso ai dati fornendo una minima appropriata autorizzazione.

Il nuovo livello di autorizzazione SECADM fornisce l'autorità per gestire l'accesso alle tabelle vietando la creazione, la cancellazione o la modifica di qualsiasi accesso alle tabelle. Il livello di autorizzazione DBADM fornisce un'opzione per avere funzionalità di amministrazione senza l'accesso ai dati. Questi profili di autorità consentono una migliore separazione dei compiti, limitando le autorizzazioni su tutti gli aspetti di una tabella e relativi dati.

Inoltre, il DB2 10 nell'ambito dell'audit, fornisce una serie di strumenti per verificare abusi e sovrapposizioni di autorità all'interno del sistema. Questo aiuta la gestione, e gli amministratori possono configurare rapidamente politiche di audit sicurezza e accesso ai dati per qualsiasi ruolo o utente.

3.7 Support for row and column access control

Il DB2 10 migliora anche la sicurezza attraverso il controllo degli accessi di righe e colonne. Questo controllo degli accessi consente agli amministratori di abilitare la protezione in una colonna o in una specifica riga particolare nella banca dati. Questa sicurezza ha lo scopo di limitare i dati visibili da qualsiasi utente finale, non consentendo che il contenuto di una colonna o di una riga sia restituito nella risposta alla query dell'utente a meno che non abbia una specifica autorizzazione. I ruoli della sicurezza combinati con label row (LBAC) e column access control (RCAC), provvedono a mascherare oppure crittografare informazioni sensibili rendendo più sicuro l'ambiente di database per il nostro business. Tutte queste caratteristiche offrono controlli più severi, consentono una maggiore flessibilità di sicurezza, e forniscono enormi capacità di audit.

4. Sezione IV - Application Enablement

4.1 pureXML enhancements

Nel DB2 10, XML può essere usato quasi ovunque, in variabili SQL, scalar functions, SQL table functions e SQL procedures. Il DB2 10 pureXML incorpora molti nuovi miglioramenti volti a migliorare le prestazioni complessive dell'XML, fornisce una facile gestione dello schema XML e abbraccia la compatibilità con la famiglia DB2.

Questi miglioramenti iniziano con la convalida dello schema XML che ora è costruito in DB2 10. Non è più necessario specificare lo schema XML perché il DB2 gestisce la convalida dello schema XML più facilmente attraverso una funzione built-in che convalida gli schemi XML. Il DB2 utilizza il timestamp per abbinare il documento XML alla corretta versione dello schema. Questo permette a più versioni dello schema di coesistere e di convalidare nuovi o vecchi documenti XML con la versione appropriata dello schema XML.

Ulteriori miglioramenti delle funzionalità offrono la possibilità di manipolare qualsiasi parte di un documento XML. Utilizzando le istruzioni SQL con espressioni XML, ogni singolo o più nodi del documento XML possono essere inseriti, aggiornati, cancellati o possono avere i loro dati aggiornati. Questo fornisce enormi funzionalità per i documenti XML, prestazioni complessive e flessibilità per qualsiasi processo applicativo.

Un nuovo XML type modifier è disponibile con il DB2 10. XML type modifier rafforza e convalida i dati della colonna del documento XML con le informazioni dello schema. Questo nuovo XML type modifier è utilizzato per aggiungere o rimuovere gli schemi XML e possono essere modificati su schemi XML precedenti così i tipi di colonna XML possono essere convalidati. Questo aiuta a garantire che lo schema dei documenti XML memorizzi solo il contenuto XML desiderato.

4.2 Support for XML date and time

Il DB2 10 ha ampliato le scelte per i data types DATE e TIMESTAMP con TIMEZONE. Questi tipi di dati sono supportati all'interno degli indici XML e il timestamp è stato ampliato per avere una maggiore precisione nella gestione dei dati.

4.3 Binary XML Support

DB2 10 migliora l'XML con il nuovo supporto degli oggetti XML binari. Il supporto binario è importante perché tale formato migliora l'interazione tra server e applicazione.

Il supporto binario utilizza il formato pre-tokenized e length definitions, che migliorano le prestazioni complessive di oggetti XML binari e forniscono una facilità d'uso per la definizione di applicazioni.

L'XML binario ha anche la caratteristica di flessibilità aggiuntive come lo String IDs per un testo che rappresenta alcune o tutte le occorrenze dello stesso testo con un identificatore integer. Questo contribuisce a limitare le dimensioni di XML. Lavorare con lo stesso testo migliora le prestazioni di un'applicazione.

4.4 Faster streaming of XML and LOBs

Il DB2 10 fornisce un nuovo parametro di installazione `LOB_INLINE_LENGTH` che imposta il numero predefinito di byte per la memorizzazione di un LOB in linea. Avendo una lunghezza LOB

minimizzata o una lunghezza standard predefinita, questo parametro fornisce una migliore capacità di streaming e la capacità di minimizzare e ottimizzare l'utilizzo dello spazio LOB in linea.

La minimizzazione delle dimensioni LOB o l'eliminazione di un LOB o materializzazione XML riduce il consumo di memoria e migliora la CPU per le operazioni dei LOB. Ciò è particolarmente utile durante la utility LOAD LOB che utilizza un file di riferimento variabile per i dati LOB, inserendo LOB o dati XML da applicazioni server DRDA remote e quando si inserisce un singolo LOB o valore XML in una riga. Utilizzando la nuova opzione LOB_INLINE_LENGTH in tutti questi casi, si riduce il numero di byte necessari e si semplificano XML e le operazioni di LOB.

4.5 XML define (NO) for LOB and XML TS

I DBA ora hanno la possibilità di ritardare la definizione del LOB o data set XML e dei loro indici. Questo aiuta a risparmiare spazio di archiviazione definendoli nel catalogo DB2 e lasciando che le applicazioni li richiamino con SELECT e FETCH. I LOB, i data set XML e i loro indici sono allocati solo quando viene effettuato il primo inserimento, risparmiando memoria e prestazioni applicative fino a quando i dati non sono veramente salvati nella banca dati.

Il DBA può anche utilizzare l'utility CHECK DATA per verificare la coerenza tra lo schema XML, i dati del documento e i dati dell'indice XML.

4.6 Temporal queries and their business advantages

Il DB2 10 fornisce nuove funzionalità di dati temporali, utilizzando le nuove definizioni di tabella BUSINESS_TIME e SYSTEM_TIME. La gestione del periodo temporale è automatica, quando i criteri temporali sono soddisfatti, i dati sono archiviati in una tabella storica associata.

Il PERIOD SYSTEM_TIME o PERIOD BUSINESS_TIME sono due colonne che definiscono il periodo temporale di permanenza dei dati all'interno della tabella. Il PERIOD SYSTEM_TIME si riferisce al tempo in cui i dati sono stati inseriti nel sistema. Il PERIOD BUSINESS_TIME riguarda il periodo in cui la transazione commerciale o di business è rilevante. Queste definizioni controllano i criteri per i quali i dati devono esistere in tabella e quando gli stessi dati devono essere migrati in una tabella storica associata. Si possono usare entrambe le definizioni in una tabella, in questo caso la tabella utilizza criteri bi-temporali per controllare la permanenza dei dati in tabella.

Con il nuovo parametro BUSINESS_TIME WITHOUT OVERLAPS nella definizione delle tabelle temporali è possibile visualizzare tutte le transazioni con un timestamp univoco. Ciò avviene usando il nuovo TIMESTAMP con precisione al picosecondo (precisione 12) che rende univoca qualsiasi operazione su tutta la tabella temporale. Si tratta di un grande vantaggio per i sistemi globali che hanno bisogno di un timestamp univoco per le transazioni commerciali.

Quando un SQL viene eseguito sulla tabella temporale, la chiave "FOR (SYSTEM_TIME o BUSINESS_TIME) DA columnA a ColumnB" specificata come predicato WHERE o altro predicato simile, può essere utilizzato in SQL per ottenere solo i dati che sono all'interno del periodo di tempo temporale di interesse. Questo aiuta e complica l'elaborazione dell'SQL come INSERT o DELETE su tabelle temporali, in quanto può accadere che più righe possono essere inserite o modificate nella tabella.

Nel caso in cui una riga rientri solo parzialmente nel periodo specificato (valore iniziale e valore finale del BUSINESS_TIME) nel predicato WHERE per un'istruzione di SQL DELETE, la riga viene eliminata e vengono inserite uno o due righe aggiuntive. Le righe inserite rappresentano i valori di riga originali per i periodi non cancellati dall'operazione di DELETE. Per le nuove righe inserite, il valore iniziale e il valore finale per il BUSINESS_TIME sono impostati in modo tale da riflettere i dati non interessati dalla istruzione di SQL DELETE.

Esempio: creare una tabella, policy_info, che utilizza un periodo SYSTEM_TIME e creare una tabella di storia, hist_policy_info. Poi effettuare un ALTER TABLE per associare la tabella policy_info con la tabella hist_policy_info (di seguito esempi da manuale di IBM SQL Reference).

```
CREATE TABLE policy_info
(policy_id CHAR(10) NOT NULL,
coverage INT NOT NULL,
sys_start TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN,
sys_end TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END,
create_id TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID,
PERIOD SYSTEM_TIME(sys_start,sys_end));
```

```
CREATE TABLE hist_policy_info
(policy_id CHAR(10) NOT NULL,
coverage INT NOT NULL,
sys_start TIMESTAMP(12) NOT NULL,
sys_end TIMESTAMP(12) NOT NULL,
create_id TIMESTAMP(12));
ALTER TABLE policy_info
ADD VERSIONING USE HISTORY TABLE hist_policy_info;
```

Esempio: creare una tabella, policy_info, che usa il periodo BUSINESS_TIME.

```
CREATE TABLE policy_info
(policy_id CHAR(4) NOT NULL,
coverage INT NOT NULL,
bus_start DATE NOT NULL,
bus_end DATE NOT NULL,
PERIOD BUSINESS_TIME(bus_start, bus_end));
```

Esempio: creare una tabella, policy_info, che usa sia il periodo SYSTEM_TIME che BUSINESS_TIME per mantenere lo storia delle righe e tenere traccia di un periodo di tempo specificato dall'utente. Una tabella che specifica sia SYSTEM_TIME che BUSINESS_TIME si definisce table bi-temporal. Per consentire la conservazione delle righe storiche, in una tabella storica, è necessario creare e associare una tabella storica (utilizzando l'istruzione ALTER TABLE) con la tabella policy_info.

```
CREATE TABLE policy_info
(policy_id CHAR(4) NOT NULL,
coverage INT NOT NULL,
bus_start DATE NOT NULL,
bus_end DATE NOT NULL,
sys_start TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN,
sys_end TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END,
create_id TIMESTAMP(12) GENERATED ALWAYS AS TRANSACTION START ID,
PERIOD BUSINESS_TIME(bus_start, bus_end),
PERIOD SYSTEM_TIME(sys_start, sys_end));
```

```
CREATE TABLE hist_policy_info
(policy_id CHAR(4) NOT NULL,
coverage INT NOT NULL,
bus_start DATE NOT NULL,
bus_end DATE NOT NULL,
sys_start TIMESTAMP(12) NOT NULL,
sys_end TIMESTAMP(12) NOT NULL,
create_id TIMESTAMP(12));
```

```
ALTER TABLE policy_info
ADD VERSIONING USE HISTORY TABLE hist_policy_info;
```

4.7 Timestamp, TIME_ZONE and other data type enhancements

Maggiore precisione del timestamp per le applicazioni Java.

Il DB2 10 migliora il tipo di dato TIMESTAMP con una maggiore precisione e capacità di essere sensibile al fuso orario (TIMEZONE), fornendo una maggiore compatibilità e funzionalità per tutti i tipi di applicazioni.

La valorizzazione di precisione TIMESTAMP supporta fino a 12 cifre di frazioni di secondo (picosecondi) con il default corrisponde al valore predefinito di Java 6 cifre di precisione di frazioni di secondo. Le 6 cifre predefinite aiutano anche la funzionalità e la compatibilità Java e Famiglia DB2 insieme con la compatibilità di SQL Server. Il CURRENT_TIMESTAMP utilizza uno speciale registro così le applicazioni possono specificare il tipo di precisione desiderato per le loro esigenze applicative. La precisione dei secondi del timestamp può essere regolata a 0 o 3 cifre se questo soddisfa i requisiti delle applicazioni.

4.8 Support for TIMESTAMP with TIME ZONE

Il TIMESTAMP con TIMEZONE è un nuovo tipo di dato DB2. Questo nuovo tipo di dato incorpora il nuovo timestamp a 12 cifre e anche il nuovo standard industriale UTC (tempo universale coordinato), che sostituisce il vecchio GMT (Greenwich Mean Time). Questo fornisce alle applicazioni un'altra capacità di fuso orario per potersi confrontare in tutto il mondo con la stessa esatta Timeline, che è vitale per vari settori come quello globale finanziario, vendite e sistema bancario.

4.9 Access to currently committed data

Con la possibilità di avere da 5 a 10 volte più thread in esecuzione contemporaneamente in un singolo membro DB2, il DB2 10 ha sviluppato una maggiore quantità di miglioramenti sulla concorrenza delle applicazioni. Il DB2 10 ora fornisce una possibilità a livello DB2 BIND package che consente di scegliere il modo in cui le applicazioni devono gestire situazioni di concorrenza sui dati.

Il nuovo parametro è il CURRENTACCESSRESOLUTION con le impostazioni USECURRENTLYCOMMITTED e WAITFOROUTCOME. Queste impostazioni sovrascrivono i parametri del sottosistema DB2, EVALUNC e SKIPUNCI e aiutano il pacchetto applicativo a eseguire rapidamente l'azione di concorrenza desiderata.

Il parametro USECURRENTLYCOMMITTED indica al sistema di ignorare le righe che sono in procinto di essere inserite e considera solo le righe commesse. Questa clausola è subordinata all'impostazione del parametro di BIND – ISOLATION LEVEL – che può essere Cursor Stability o Read Stability.

Il parametro WAITFOROUTCOME segnala al sistema di attendere le righe che sono “in volo” che possono essere risolte da un rollback o commit. Questo fa sì che l'applicazione aspetti per l'inserimento o elimini le attività di commit o rollback prima di determinare se le righe possono essere incluse nella risposta SQL all'applicazione.

Queste diverse impostazioni garantiscono all'applicazione la flessibilità necessaria per gestire le transazioni web molto concorrenti, attendere o utilizzare i dati non impegnati e dare una grande flessibilità per l'architettura aziendale. Queste diverse impostazioni dei parametri contribuiscono a fornire il livello di concorrenza per il pacchetto desiderato e inoltre forniscono funzionalità che imitano le impostazioni di concorrenza per le applicazioni di un altro fornitore di database.

4.10 SQL compatibility improvement - Extended indicator variable

Il DB2 10 introduce una nuova **variabile indicatore estesa** che fornisce un modo per specificare che non vi è alcun valore in una colonna per le istruzioni di INSERT, UPDATE e MERGE. Ad esempio se la **variabile indicatore estesa** abilitata contiene -5, specifica il valore predefinito. Se la **variabile indicatore estesa** non è abilitata, il -5 specifica un valore NULL. Se, la variabile indicatore estesa è abilitata e il valore è -7, indica che la variabile deve essere UNASSIGNED, ignorata e trattata come se non esistesse all'interno dell'istruzione SQL.

Queste variabili indicatore estese, sono in genere utili per le applicazioni Java soprattutto in quelle applicazioni che utilizzano istruzioni dinamiche e istruzioni SQL variabili, dove il numero di possibili variabili host è sconosciuto fino a quando la logica transazionale è completata e il codice SQL deve essere eseguito.

Questo risolve il problema per le applicazioni che in precedenza richiedevano la codifica di molte istruzioni SQL in base ai valori disponibili per le istruzioni SQL. Ora queste molteplici istruzioni SQL possono essere consolidate. Quando il valore della colonna non è noto, il valore della variabile host può utilizzare la nuova parola chiave UNASSIGNED per la colonna appropriata.

Ciò è particolarmente importante per le applicazioni che utilizzano istruzioni SQL dinamiche che intasano la Dynamic Statement Cache del sistema con molte copie essenzialmente della stessa istruzione SQL.

4.11 Extended support for implicit casting

Il cast implicito è l'automatica conversione di diversi tipi di dati con un altro compatibile. Il DB2 esalta il suo cast implicito mediante la gestione di tipi di dati numerici che sono in grado di essere implicitamente convertiti in tipi di dati carattere o stringa grafica. Supporta la conversione dei dati nella direzione opposta dal tipo di dato carattere o stringa grafica verso un tipo di dati numerici.

Nelle versioni precedenti di DB2, questo doveva essere fatto manualmente. Ora i dati numerici, stringa di caratteri e stringa grafica possono essere trattati, confrontati e assegnati implicitamente. DB2 per z/OS migliora la compatibilità e la portabilità dell'SQL da altri sistemi di database.

4.12 Enhanced scalar function support

Il DB2 10 migliora la compatibilità con altri fornitori di database in merito all'SQL scalar e alle table functions. Queste built-in functions sono utilizzate in tutte le applicazioni SQL per velocizzare le funzioni OLAP, e calcoli come SUM, AVG, SIN, COS e molti altri. Come nelle precedenti release, queste funzioni in linea utilizzate in istruzioni SQL restituiscono un singolo valore.

Altre funzioni SQL scalar non-in-linea contengono la logica, forniscono funzionalità aggiuntive e la flessibilità di un'applicazione. Questa flessibilità consente la compatibilità della famiglia DB2 e la migrazioni di dati di altri fornitori di database. Il DB2 supporta anche più versioni e gestione del codice sorgente di queste funzioni in base alla loro lista di parametri, opzioni delle routine e il corpo della funzione. Queste funzioni possono essere modificate, sostituite con versioni diverse distribuite su più server per aiutare la fase di test e le prestazioni complessive. Il fallback a una versione precedente viene fatto immediatamente senza necessità di Rebind o ricompilazione.

Il DB2 10 introduce il supporto di SQL user-defined table functions che aiuta ad alleviare la migrazione di applicazioni di altri fornitori di database. Le table functions sono molto flessibili poiché restituiscono un unico risultato basato sul numero di diversi tipi di parametri come i LOBs, distinct types, tabelle e tabelle di transizione.

4.13 SQL procedural language enhancements

Il DB2 9 ha fornito nuovo supporto nativo per il linguaggio procedurale SQL, eliminando il requisito ingombrante di generare un programma C dalla procedura SQL che doveva poi essere eseguita come una stored procedure esterna. Le procedure SQL possono essere eseguite nativamente all'interno del motore DB2 per l'esecuzione runtime e memorizzate nel catalogo DB2 Catalogo per una gestione migliore e il controllo della versione. L'esecuzione del codice nativo all'interno del motore DB2 aiuta anche il debug, la distribuzione e la gestione di versioni di procedure SQL tra più server. Memorizzare la procedure SQL migliora il controllo generale e il cambiamento di questo codice in modo che possa essere gestito come gli altri moduli sviluppati per l'applicazione.

Il linguaggio procedurale SQL presenta molti miglioramenti come le SQL table functions, istruzioni SQL composite nidificate all'interno di una procedura e l'istruzione RETURN che può restituire il result set di un'istruzione SQL SELECT. Queste Stored Procedure e gli altri miglioramenti del linguaggio procedurale SQL consentono ogni tipo di lavorazione.

I miglioramenti delle procedure SQL previsti col DB2 10, forniscono la compatibilità necessaria con altri fornitori di database. Il linguaggio procedurale offre una maggiore capacità di accettare molti tipi di dati e XML come parametri e fornire un limitato uso di cursori scrollabili e grande compatibilità e opportunità di integrazione tra le applicazioni. Tutti questi miglioramenti, offrono l'opportunità di migrare verso altre soluzioni di gestione per i database relazionali per l'ambiente z/OS per una migliore esperienza del costo di possesso e dalle prestazioni uniche, disponibilità, scalabilità e capacità che possono essere trovate solo con DB2 per z/OS e System z.

5. Sezione V – Data Warehousing

5.1 Support for temporal tables and versioning

Il DB2 10 ha molte caratteristiche e funzioni che lo rendono una piattaforma ideale per applicazioni di data warehousing. I miglioramenti, come la gestione dei dati temporali, automatic SQL optimizer, nuovo Hash access plan, SQL scalar function e table function, nuovo timestamp con precisione ai nanosecondi con nessuna sovrapposizione di precisione, timestamp con fuso orario, disponibilità e scalabilità hanno contribuito alla sua superiorità per il data warehousing e per le applicazioni di business intelligence.

Il miglioramento del temporal data é un importante passo in avanti, che consente di archiviare automaticamente le versioni di dati in tabelle storiche. Quest'archiviazione automatica combinata con un tablespace partizionato che espandendosi automaticamente consente di archiviare i dati quando vengono memorizzati nel sistema.

Molte novità dell'SQL optimizer migliorano le prestazioni per molte query ad-hoc inviate da molti utenti simultaneamente tramite tools OLAP di terze parti o da Excel attraverso connessioni ODBC. Queste query ad-hoc con i loro criteri di BUSINESS TIME o SYSTEM TIME potranno gestire correttamente l'aspetto bi-temporal dei dati del data warehouse e restituire a fronte di richieste SQL le risposte desiderate agli utenti finali.

Il nuovo Hash Access plan riduce il tempo di recupero di informazioni verso il basso per un singolo I/O con tempi di risposta eccezionali.

Per tutte queste caratteristiche e funzionalità unite ad una elevata disponibilità, più scalabilità e affidabilità, System z, offre una piattaforma di data warehouse per prestazioni di sistema ottimali e sistemi di business intelligence operativi trasmessi all'intera impresa.

5.2 OLAP functionality built directly into DB2

Supporto per OLAP - moving sums, averages, and aggregates .

Le funzionalità OLAP di moving sums, averages e aggregates ora sono costruite nel DB2. Miglioramento dell'SQL, i risultati intermedi nei workfile, SQL scalar function e table function forniscono miglioramenti prestazionali per le attività OLAP.

Moving sums, averages, e aggregates sono funzionalità comuni OLAP all'interno di qualsiasi applicazione di data warehousing. Sono calcoli standard tipici che sono fatti utilizzando diversi tipi di raggruppamento: periodo-temporale oppure localizzazione-vendita di prodotti oppure la posizione dei negozi oppure altri criteri di localizzazione.

Essendo queste funzionalità OLAP costruite direttamente in DB2, consentono una elaborazione SQL standard, applicazioni ripetibili, SQL function o table functions e prestazioni robuste grazie a migliori processi di ottimizzazione.

Queste funzionalità OLAP sono ulteriormente migliorate attraverso SQL scalar, table functions e nuove tabelle temporali per stabilire una finestra dati per la somma mobile, la media o aggregato per calcolare il set di risposta. Utilizzando una partizione, il periodo di tempo o una common table expression (CTE) le funzioni OLAP sono in grado di fornire calcoli standard complessi o semplici richiesti dal data warehouse.

Visti anche i miglioramenti con SQL queste moving sum, averages e aggregates possono essere inclusi in espressioni, SELECT list o nell'istruzione ORDER BY, soddisfacendo qualsiasi esigenza applicativa.

5.3 Advanced business analytics

Le funzionalità standard disponibili in DB2 10 forniscono analisi di business avanzate. Built-in pureXML, LOB e SQL scalar e table functions offrono molte funzioni e capacità che possono essere estese a qualsiasi tipo di funzionalità personalizzata per le esigenze di business.

Le tabelle temporali e la loro capacità di archiviare automaticamente i dati storici forniscono una gestione attiva dei dati migliorando le prestazioni e l'audit. Con SQL si possono usare i parametri SQL di BUSINESS TIME e SYSTEM TIME per qualificare la risposta.

6. Sezione VI – Reduced Total Cost of Ownership (TCO)

Il DB2 10 riduce la richiesta totale di CPU fino al 5-20% in diversi modi, quando tutti i nuovi miglioramenti sono pienamente sfruttati. In molti casi la riduzione della CPU è integrata direttamente in DB2 e non richiede modifiche alle applicazioni. Alcuni miglioramenti sono attuati attraverso le normali attività quotidiane DB2 come il REBIND, la ristrutturazione delle definizioni di database, il miglioramento delle applicazioni, e le utilities. Tutte queste caratteristiche di riduzione della domanda di CPU hanno il potenziale di fornire un significativo risparmio del costo totale di possesso.

Miglioramenti nella ottimizzazione riducono i costi per l'elaborazione SQL attraverso access-path più efficaci. Miglioramenti attraverso il metodo di accesso range-list-index scan, in list pre-fetch, più parallelismo per i processi di SELECT e INSERT, miglior utilizzo dell'uso dei workfile, migliore gestione del RID pool, valutazione dell'access-path e miglioramento del flusso dei messaggi transazionali in DDF, forniscono più efficienza senza apportare alcuna modifica alle applicazioni. Tutti questi miglioramenti riducono il consumo totale di CPU a causa di una migliore efficienza del nuovo DB2 10 for z/OS.

Altri miglioramenti richiedono nuove definizioni di database o tecniche di programmazione di applicazioni per ridurre i costi complessivi. Questi miglioramenti come ad esempio l'HASH access-path, includere colonne in un indice univoco, consolidare l'SQL utilizzando la nuova funzione Attributes, e utilizzare le automatic data statistic riducono il costo della CPU attraverso un miglioramento dell'access-path.

La riduzione dei costi ridotti può essere ottenuta anche attraverso una migliore gestione e una migliore disponibilità del DB2 10. Il DB2 10 con una migliore gestione della memoria, la capacità di gestire da 5 a 10 volte più utenti, la capacità di saltare le righe bloccate, miglioramenti nel catalogo DB2, più possibilità di modificare gli schema on-line e più utilities on-line, elimina i tempi di inattività del sistema. Il DB2 10 mantiene l'applicazione disponibile anche quando più utenti, applicazioni, database, modifiche e utilities sono in esecuzione nel sistema. Processi più automatizzati aiutano a installare, configurare e semplificare l'installazione del nuovo sistema DB2 10. Uno step di installazione di pre-migrazione aiuta a catturare le impostazioni esistenti e a fornire le impostazioni appropriate per le nuove routine e programmi DB2. Queste nuove procedure hanno lo scopo di ridurre il tempo per installazione, configurazione e test.

La possibilità di migrare direttamente dal DB2 V8 permette a chi è rimasto indietro nelle versioni di software di saltare l'implementazione del DB2 9. La migrazione diretta dalla versione 8 alla versione 10 offre un modo rapido per sfruttare immediatamente tutte le grandi novità inserite nel DB2 9 (vedi appendice).

Il DB2 10 offre un valore straordinario, riducendo i costi operativi e di manutenzione, e migliorando le prestazioni. Prevedere di installare la versione DB2 10 il più presto possibile, permette di utilizzare tutte queste enormi migliorie.

7. Appendice

DB2 9 for z/OS Enhancement List

APPEND option at insert	Native SQL Stored Procedures, able to use zIIP
Automatic creation of database objects	Network Trusted Context
Autonomic index page split	OLAP Zparms new options
Autonomic re-optimization	ORDER BY and FETCH FIRST n ROWS in sub-select and full-select
Buffer management by WLM	ORDER OF extension to ORDER BY
CLONE Table: fast replacement of one table with another	On-line CHECK DATA and CHECK LOB
Change SCHEMA & VCAT	On-line REBUILD INDEX
Conditional restart: automatic search for appropriate checkpoint	On-line RENAME COLUMN
DECIMAL FLOAT, BIGINT enhancements	On-line RENAME INDEX
Database ROLES	Optimization Service Center
Enhanced CURRENT SCHEMA	Partition-by-growth table spaces
Enhanced STOGROUP definition	Preserving consistency when recovering individual objects to a prior point in time
FETCH CONTINUE	RLF improvements for remote application servers such as SAP
Faster REORG by intra-REORG parallelism	Recovery of individual table spaces and indexes from volume-level backups
Faster and more automatic DB2 restart	SELECT FROM UPDATE/DELETE/MERGE
Faster operations for variable length rows	SHRLEVEL(REFERENCE) for REORG of LOB table spaces
Generalizing sparse index and in-memory data caching method	Skipping locked rows Support for optimistic locking
Global query optimization	TRUNCATE TABLE statement
IPv6 support	Tape support for BACKUP and RESTORE SYSTEM utilities
Index Lookaside enhancements	Temporary space consolidation
Index on expressions	Unified Debugger
Index page sizes 8K, 16K, 32K	Universal Table spaces
LOB Lock reduction	Utilities CPU reduction
LOBs Network Flow Optimization	Utilities improvements
Logging enhancements	Utility TEMPLATE switching
MERGE statement	VARBINARY, BINARY
MODIFY RECOVERY enhancements	Various scalar functions
Modify early code without requiring an IPL	XML support in DB2 engine
More on-line REORG by eliminating BUILD2 phase	
NOT LOGGED table spaces	

8. Riferimenti

8.1 Articoli

How to Leverage DB2's Automated Time Travel Queries and Temporal Tables by Daniel L. Luksetich in Enterprise TechJournal

8.2 Presentazioni

A new Catalog and Directory Structure in DB2 for z/OS by Ken McDonald, Steven Thomas
DB2 10 for z/OS Performance Enhancements by James Guo
DB2 10 for z/OS: Temporal Tables (aka Time Travel Query) by Charles Lewis

8.3 Redbooks

DB2 10 for z/OS Performance Topics
Security Functions of IBM DB2 10 for z/OS
DB2 10 for z/OS Technical Overview