

Esperienze Reali Migrazione alla V10

Chi Siamo?

- Ambiente Bancario
- Gruppo Sistemisti DB2
 - 2 Persone
- Sottosistemi Gestiti
 - 15 di TEST/Sviluppo.
 - 8 di produzione
 - 5 datasharing di test
 - 1 datasharing di produzione a 3 vie

Perché Migrare alla Versione 10?

- Funzionalità Particolare della V10?
 - Risparmio CPU
- Fine servizio V8?
 - Perché alla V10 e non alla V9?
 - Risparmio CPU
- Spinti dal Management
 - Promozioni Contrattuali
 - Out of Currency

Piano Della Migrazione

- Conversion Mode
 - Test
 - *Agosto 2011*
 - QA/Collaudato
 - *Dicembre 2011*
 - Produzione
 - *Aprile 2012*
- NFM
 - Test
 - *N/A*
 - QA/Collaudato
 - *N/A*
 - Produzione
 - *N/A*

Dove Siamo Oggi...

- Ad oggi abbiamo:
 - **10** sottosistemi in **V10 Conversion Mode(V8)**
 - da **almeno 3 mesi**
 - Tutti Sistemi di **TEST**
 - **0** Sottosistemi in **NFM**
 - Ritardo rispetto al piano di 2 mesi
 - Perché ?? Vedere slide successive

Problemi Conosciuti Riscontrati

- Da settembre 2011 ad oggi applicati circa **300 PTF**
- Funzione **CHAR / VARCHAR** di un campo **DECIMAL**
 - Usermod **AN50639** provata in collaudo; non installata in produzione
 - Processo di estrazione complesso
 - scrittura record smf, lettura smf, carico in tabella db2 etc...
 - Solo per la funzione VARCHAR e CAST !!!!
 - Informazione troppo generica su statement dinamici
 - plan distserv, ipaddress generico
 - Tramite prodotto terze parti
 - estrapolati programma statici contenenti le built-in function evidenziate
 - eseguita analisi manuale e a campione su statement dinamici (da noi molto utilizzati)
 - Individuate circa **350 occorrenze** da sanare!!
 - Inviata comunicazione a applicativi
 - sui riscontri trovati da analizzare e sanare
 - su nuove linee guida per utilizzo di queste built-in function
- Incompatibilità **Prodotti Third Party ?**
 - Migrata suite CA R15 (detector/fastunload/fastload etc..)
 - Migrata BMC V10 (System performance + Opertune)

Problemi “Scoperti”

- TEMPDB / DSNDB07
 - V8 – DSNDB07 per Workfile & le CGTT
 - Tablespace Semplici e User Defined
 - V8 – Database “TEMP” dedicato alle DGTT
 - Tablespace Segmentati e DB2 Defined
 - **SEGSIZE = 4** con possibilità di modificarlo
 - V9/V10 – Database “TEMP” eliminato, le DGTT usano il DSNDB07
 - TS Originali rimangono TS Semplici, User Defined
 - Nuovi TS definiti Segmentati e DB2 Defined
 - **SEGSIZE = 16 e non può essere modificato durante la CM !!!**
 - Nuovo ZPARM WFDBSEP per separare DGTT in TS dedicati
 - In base al DB2-Defined e alla SECQTY > 0 (or -1)
 - V9/V10 – DB2 utilizza le 32K per DGTT con RECL > 100byte
 - Dove è il problema ??

Problemi “Scoperti”

- TEMPDB / DSNDB07 cont.
 - Programma BATCH utilizza una DGTT con RECL = **250**
 - **Ripetutamente** Inserisce poche righe(< 10), le legge, poi le cancella con **MASS DELETE**
 - Non effettua COMMIT
 - V8 – apparentemente nessun problema, spazio allocato al TEMP DB sufficiente
 - **7 TS di 1000 traccie** ognuno
 - V10 CM – DSNDB07 esplode!!!
 - Definiti **4 TS** di **2GB** ognuno per le 32K durante la migrazione
 - Più 6 TS per le 4K (2 x2GB e 4 x 2000 traccie)
 - Durante esecuzione del Batch, il DB si è esploso
 - Per far girare il Batch sono stati definiti **10 TS** di **2GB** ognuno per le 32K
- Perché ?? ...

Problemi “Scoperti”

- TEMPDB / DSNDB07 cont.
 - Programma fa Mass Delete
 - La DGTT è allocato in un TS Segmentato
 - Tutto il Segmento segnato come DELETED
 - Spazio inutilizzabile senza una COMMIT
 - INSERT successivi vanno in un SEGMENT nuovo
 - V8 utilizza Pagesize 4K con Segsize 4
 - Quindi 16K di spazio non utilizzato per ogni loop di INSERT / DELETE
 - V10 CM utilizza Pagesize 32K con Segsize 16
 - Quindi 512K utilizzato per ogni loop di INSERT / DELETE
 - Ovvero 32 volte quello della V8!!!!
 - **Quindi, 10mila loop di INSERT / DELETE di 2 righe**
 - **Serve 5GB di spazio disco!!**

Problemi “Scoperti”

- TEMPDB / DSNDB07 – Soluzioni
 - Inserire COMMIT nel programma
 - Segmento viene riutilizzato
 - Necessario intervento applicativo
 - Soluzione “a posteriore”
 - Impossibile individuare programmi che usano le DGTT in questa maniera senza fare COMMIT
 - Effettuare la DELETE WHERE 1=1
 - Ogni riga cancellata singolarmente
 - Implicazioni prestazionali notevoli
 - Necessario intervento applicativo
 - Impossibile individuare programmi che usano le DGTT in questa maniera per modificarli
 - Interventi sulla definizione del DSNDB07 e WFDBSEP = NO
 - La DGTT viene allocata comunque nel TS 32K
 - La DELETE di MASSA marca deleted comunque 16 pagine e non le riutilizza
 - Senza un TS DB2-Defined l’utilizzo della DGTT riceve -904

Problemi “Scoperti”

- TEMPDB / DSNDB07 – Soluzioni
 - Utilizzare una Created Global Temporary Table
 - Necessario intervento applicativo
 - **Richiesta Usermod a IBM per:**
 - Fare in modo che l’algoritmo di **selezione dei TS avvenga come nella V8;**
 - in pratica per righe fino a lunghezza 4K il db2 selezionerà i TS DSN4Kxx e non i DSN32Kxx.
 - Utilizzando quindi un pagesize da 4K invece che da 32K!
 - Il risparmio sarà quindi di circa 8 volte !
 - Usermod **AN37361** Applicata e funzionante !

Prossimi Passi

- Portare i Sottosistemi Produttivi a V10 CM
 - ***Primo sottosistema di produzione ieri 12-3-2012 !!!***
- Portare Sottosistemi TEST a V10 NFM
 - ***entro....?????***

THE END

Grazie !!