



TIPS & TRICKS

Il Supporto Tecnico vi svela i segreti del mestiere

In questo numero: **Scalabilità e prestazioni: ottenere il massimo dalla piattaforma SAS**

Scalabilità e prestazioni: ottenere il massimo dalla piattaforma SAS

Introduzione

Memorizzare enormi moli di dati senza impattare sulle prestazioni dei sistemi è una delle più importanti sfide per molti professionisti dell'Information Technologies.

SAS Scalable Performance Data Server consente di vincere questa sfida e di avere un database in grado di adeguarsi alle esigenze e ai cambiamenti aziendali senza impattare sulle applicazioni.

Utilizzando le più moderne tecniche di esecuzione parallela e funzionalità di data server, SAS Scalable Performance Data Server fornisce una soluzione integrata che consente ad un enorme numero di utenti concorrenti l'accesso trasparente ad una base dati di notevoli dimensioni.

SAS Scalable Performance Data Server

Il prodotto SAS SPD Server è un data server che si basa su tecnologia client-server e multi-user, disegnato per ottimizzare la memorizzazione dei dati e la velocità di accesso a data set SAS di notevoli dimensioni attraverso la parallelizzazione di molte funzioni di I/O quali, ad esempio, letture condizionate (WHERE), creazioni di indici, ordinamento implicito tramite lo statement BY, esecuzioni di GROUP BY e di SQL Passthru. SPD Server richiede un hardware SMP ed è disegnato per utilizzare tutte le risorse disponibili per ottenere la massima scalabilità. Si ottiene il massimo beneficio da SPD Server quando è eseguito su una macchina con le seguenti caratteristiche:

- CPU multiple
- Canali di I/O multipli
- Dischi multipli
- Grosse quantità di dati da partizionare

In aggiunta alla parallelizzazione, SPD Server fornisce la sicurezza dei dati attraverso la validazione di userid/password e Access Control Lists (ACL), e funzionalità di backup e recovery.

SPD Server 3.0 supporta sessioni server su HP-UX, Solaris, AIX, Compaq's Digital Unix, e Windows NT. Le applicazioni client possono essere eseguite sulla stessa macchina o su una diversa e si collegano al server attraverso l'istruzione *libname*. In questo modo, se si rendesse necessario cambiare la configurazione della componente server, l'unica modifica da apportare a livello client riguarderebbe la sola istruzione *libname*.

Componenti tecnologici

Per utilizzare SPD Server è necessario lanciare sulla macchina server due eseguibili: uno per eseguire il *name server*, l'altro per eseguire il *data server* vero e proprio. Quando un'applicazione client si collega al SPD Server attraverso una libname, viene fatta partire automaticamente una sessione SAS proxy (spdsbase) che eseguirà tutte le sue richieste (Fig. 1). Se l'applicazione client non è una applicazione SAS, e quindi viene sfruttato lo standard ODBC o JDBC, viene utilizzato un terzo processo server, *spdssnet*.

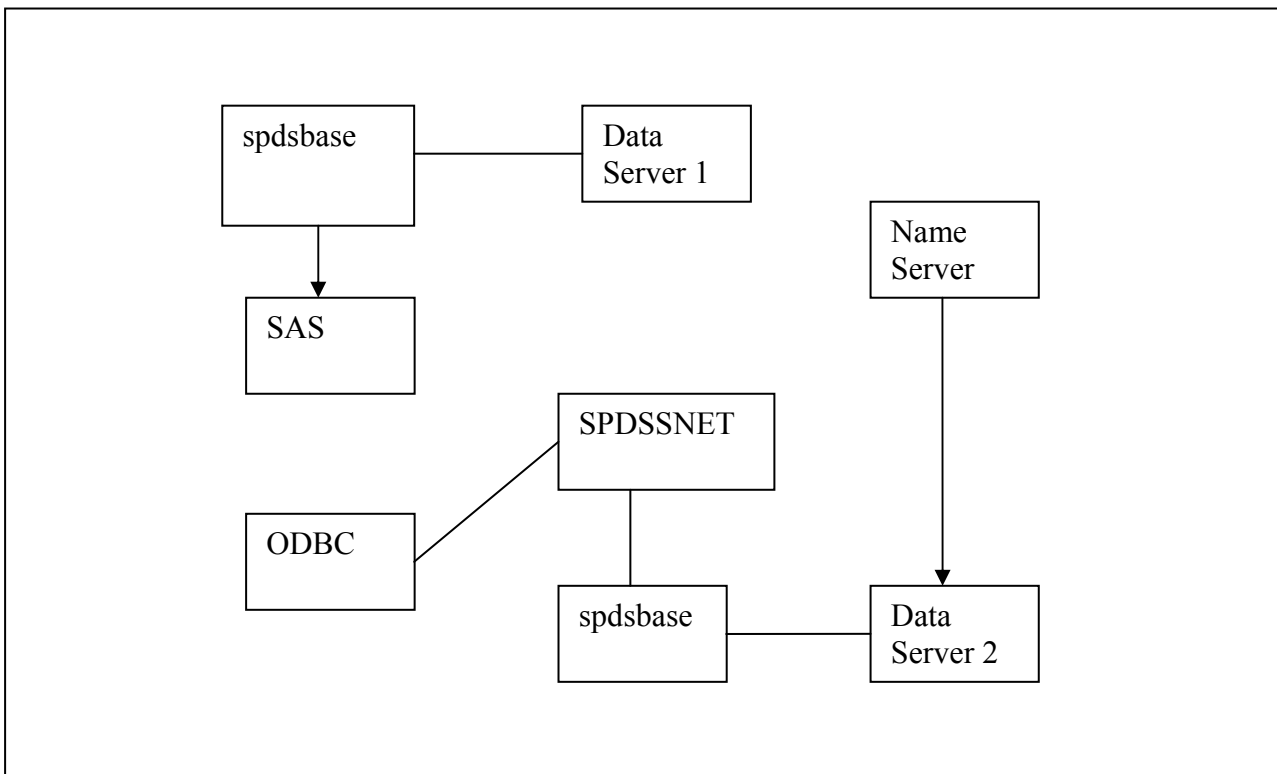


Figura 1.

Name Server (SPDSNSRV)

Il name server è il componente che gestisce la mappatura logica tra il nome del database e il DATA SERVER definito per quel dominio. Questo meccanismo consente ai programmi usati dagli utenti finali di ignorare su quale server si trovi il dominio a cui accedono. Di conseguenza, se il server viene cambiato fisicamente, i programmi non devono essere variati. Tutti i *data server* devono essere registrati nel *name server*. Per questioni di sicurezza, ogni client si deve collegare al *name server* prima di poter accedere al *data server*.

Data server (SPDSSERV)

È la componente che corrisponde al dominio di database. Essa, dopo aver validato l'utenza utilizzata dal processo client, fa partire un processo proxy (spdsbase), che gestisce tutte le richieste del client. Con questo meccanismo gli ambienti relativi a ciascun client sono separati e si realizza un migliore bilanciamento del sistema.

Proxy (SPDSBASE)

Ogni utente che si è autenticato con successo al data server viene automaticamente collegato ad un processo proxy. Questo processo esegue tutte le richieste del client quali accesso ai dati, esecuzione di query SQL, ecc... Quando il client si disconnette, il processo proxy termina e tutte le risorse ad esso collegate vengono riutilizzate dal sistema operativo. Questo modello garantisce il massimo bilanciamento e isolamento delle risorse. Il processo proxy è internamente suddiviso in thread in modo da ottenere le massime prestazioni attraverso l'uso del parallelismo e dell'esecuzione contemporanea.

ODBC/JDBC Server (SPDSSNET)

È il server che si occupa di convertire le richieste dei driver ODBC/JDBC forniti per il supporto dello standard open data nel protocollo SDP.

Applicazioni Client

Le applicazioni client che si collegano ad un SPD server possono essere sia SAS che non SAS. L'accesso da parte di sessioni SAS client avviene attraverso l'engine SASSPDS dell'istruzione libname, oppure attraverso l'istruzione sql di connect, o ancora attraverso l'engine ODBC. L'accesso da applicazioni diverse da SAS avviene attraverso il SAS ODBC Driver, il SAS JDBC Driver, oppure le librerie C di runtime fornite con SPD Server. Di seguito alcuni esempi di libname e query sql:

```
LIBNAME in SASSPDS 'tmp' SERVER=spdsnode.5127 USER='ghost' PROPMT=yes;
DATA in.foo; x=10; RUN;
```

```
PROC SQL;
    CONNECT TO SASSPDS(DBQ='tmp' SERVER=spdsnode.5127 USER='ghost'
    PROPMT=yes);
    SELECT * FROM CONNECTION TO SASSPDS (SELECT *....);
RUN;
```

Configurazioni ottimali per il sistema di I/O.

SPD Server è quindi una interfaccia di I/O di tipo multi-thread per la lettura parallela dei dati. L'esecuzione in modalità multi-thread aggiunge un certo rallentamento ma esso viene recuperato quando l'engine è utilizzata per la lettura di grosse moli di dati. È per questo motivo che è consigliabile utilizzare l'engine base di SAS quando i data set non sono molto grossi.

SPD Server utilizza 4 differenti aree per memorizzare i dati e altre informazioni:

- Area dei metadati
- Area dati
- Area indici
- Area di lavoro

Ciascuna di queste aree necessitano di diverse configurazioni hardware al fine di ottimizzare le prestazioni.

La figura 2 rappresenta la diversa architettura di SPD Server rispetto alla architettura dati dell'engine BASE di SAS.

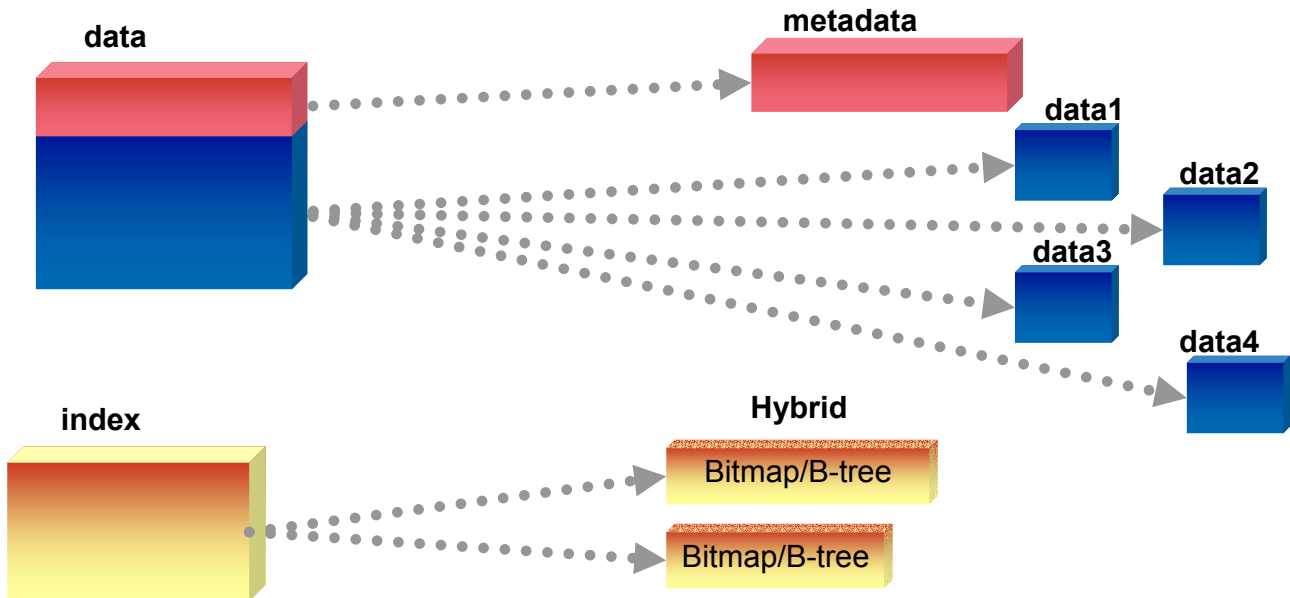


Figura 2

1. Area dei metadati

L'area dei metadati contiene informazioni relative al dominio dei dati, alle tabelle che li compongono e ai loro indici. SPD Server usa quest'area anche per memorizzare risorse non tabellari, quali ad esempio cataloghi e viste. E' di vitale importanza non perdere nessuna di queste informazioni, perciò quest'area necessita di una impostazione di dischi che prevedano un ridondanza primaria, come ad esempio il RAID1, anche nota come *mirroring*.

2. Area Dati

L'area dati corrisponde alla locazione dei file che compongono i dati veri e propri. Quest'area richiede una specifica configurazione per poter fornire un elevato throughput di I/O e poter garantire la scalabilità e disponibilità.

La locazione fisica dell'area dati viene definita dall'opzione DATAPATH dell'istruzione libname che definisce il dominio del database nel file di configurazione di SPD Server (libnames.parm).

```
libname=mydomain pathname=/spdsmeta
options="metapath=('/spdsmeta')
datapath=('/spdsdata1' '/spdsdata2' '/spdsdata3' '/spdsdata4')
indexpath=('/spdsindex1' '/spdsindex2');
```

L'opzione DATAPATH= indica una lista di *file system* (per le piattaforme unix) o di *disk drive* (per Windows) dove sono memorizzate le *table partition*. La prima table partition sarà memorizzata nel primo file system della lista, la seconda nel secondo file system e così via. Dopo che anche l'ultimo file system è stato utilizzato, la successiva partizione sarà nuovamente memorizzata nel primo file system, quindi, di conseguenza, tutti i file system saranno utilizzati approssimativamente allo stesso modo.

La configurazione del file system che contiene l'area data è cruciale per il tipo di performance ottenibili in fase di accesso ai dati attraverso SPD Server. Per consentire l'accesso parallelo ai dati, le tabelle dati sono suddivise in diversi file fisici, chiamati *data partition files (DPF-components)*, che dovrebbero essere memorizzati su dischi multipli.

Dimensionamento dei *table partition*

La dimensione della partizione dovrebbe essere scelta facendo in modo che in ciascun file system definito per il *datapath* siano memorizzate tre o quattro *table partition* per ciascuna tabella. Il numero di *table partition* per ciascun *file system* non dovrebbe essere superiore a dieci. Il principale svantaggio dell'aver troppe partizioni consiste nell'eccessivo numero di aperture di file (file open handles) necessari per la lettura della tabella, che impatta negativamente sulle risorse di sistema e sulle altre applicazioni. Inoltre, avere troppe table partition non dà alcun beneficio in termini di prestazioni. La seguente formula può essere usata come linea guida per stabilire la corretta dimensione della *table partition*:

$$\text{partition size} = (\# \text{rows} * \text{row length}) / (\# \text{data file systems} * \text{max partitions per file system})$$

La dimensione della partizione risultante da questa formula dovrebbe poi essere arrotondata a valori quali 16, 64, 128, 256MB e così via.

Configurazione dell'area dati

In un sistema multiprocessore, sarebbe opportuno cercare di avere tanti *datapath* quanti sono i processori così come sarebbe buona norma avere un controller di I/O per ciascun datapath. In relazione poi alla dimensione di banda dei dischi utilizzati, potrebbe essere necessario avere a disposizione ulteriori controller. In linea di massima, occorre impostare una configurazione il più semplice possibile, con una relazione uno-a-uno tra il numero di dischi, o gruppi di dischi in caso di RAID, e il numero di file system utilizzati.

Ad esempio, su una macchina quadriprocessore, la configurazione più semplice ed efficace sarebbe costituita da quattro dischi (o gruppi di dischi in caso di RAID) contenente ciascuno un file system, come rappresentato in figura 3.

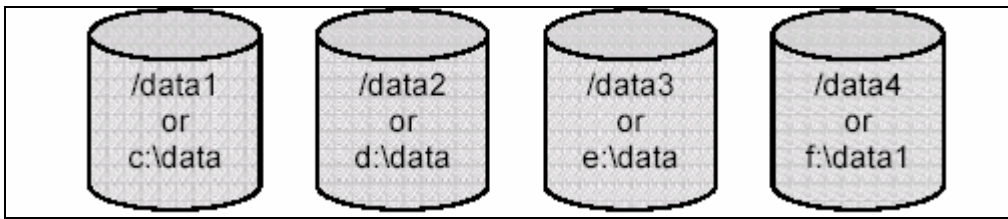


Figura 3. Quattro dischi singoli dedicati alle partizioni dell'area dati

Per ottenere le migliori prestazioni, sarebbe opportuno dedicare in maniera esclusiva a SPD Server i dischi sui quali viene memorizzata l'area dati. Inoltre, ciascuno di questi dischi potrebbe essere sostituito da un *stripe-set* di dischi in RAID 0. Maggiore è lo *striping* applicabile, migliori saranno le performance ottenibili.

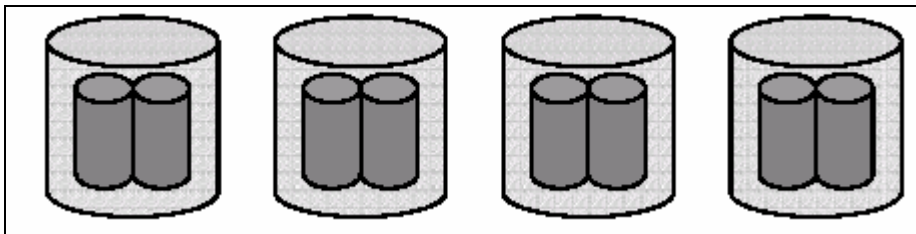


Figura 4. Quattro array di dischi, ciascuno in stripe su due dischi

Tuttavia, se uno dei dischi rappresentato in figura 4 dovesse avere un problema, tutti i dati del dominio sarebbero persi, in quanto non vi è alcun meccanismo di ridondanza nella configurazione in striping. Per poter implementare la ridondanza dei dati, ciascuno di questi array di dischi in RAID 0 dovrebbe essere sostituito con un disk array in mirroring (RAID 1), oppure un stripe-set in mirroring (RAID 10), o un array in RAID 5. La configurazione in RAID 10 è la soluzione migliore sia in termini di prestazioni che in termini di sicurezza del dato. Essa richiede almeno 4 dischi in ciascun array. La configurazione in RAID 5, nota anche con il nome di "striping with rotating error correction code" (ECC), consente il miglior rapporto tra ridondanza e performance da una lato e costi dall'altro. La configurazione minima, infatti, richiede solo tre dischi per array, come rappresentato in figura 5. Tuttavia, vi è un minimo di penalizzazione per la velocità di scrittura in quanto l'informazione relativa al ECC (error correction code) deve essere aggiornata ogni volta che vengono modificati i dati.

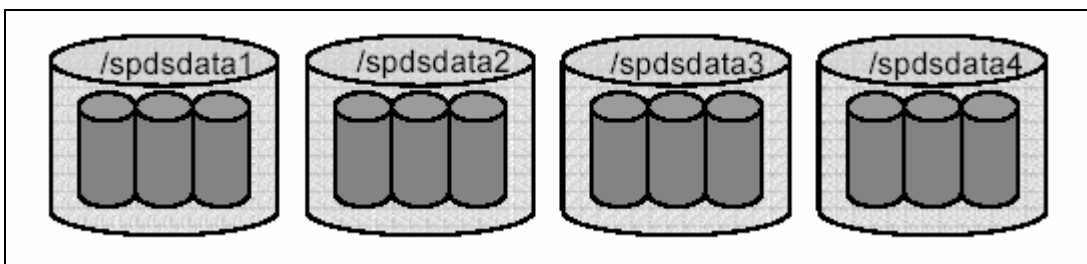


Figura 5. Quattro array di dischi in RAID 5, ciascuno in stripe su tre dischi

Solitamente, i dischi di un array sono organizzati in gruppi, ciascuno dei quali connesso con il proprio controller. La figura 6 rappresenta due disk tower contenenti otto dischi e due controller ciascuno. Quattro dischi sono raggruppati per ogni controller.

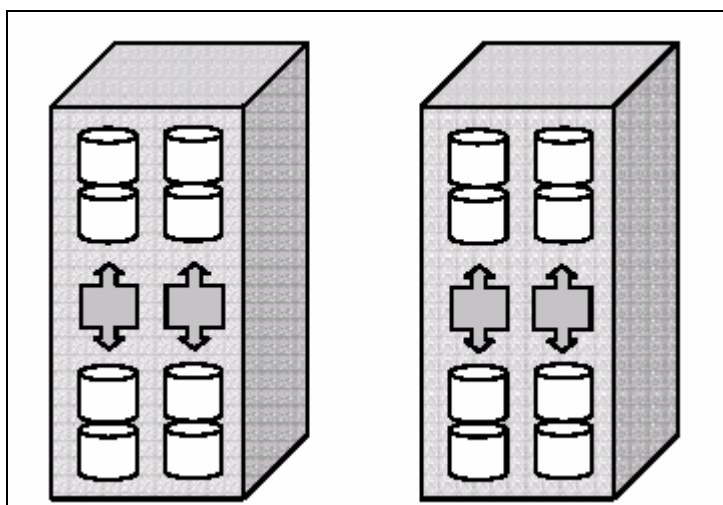


Figura 6. Due disk tower

Ipotizzando che ciascun disco abbia un throughput di 35MB/s, e ciascun controller sia connesso a due canali per un throughput di 80MB/s, due dischi potrebbero effettivamente saturare un canale del controller. Per questo motivo è consigliabile distribuire in modo omogeneo sui canali e i controllers disponibili i dischi utilizzati in configurazioni di stripe-set o mirror.

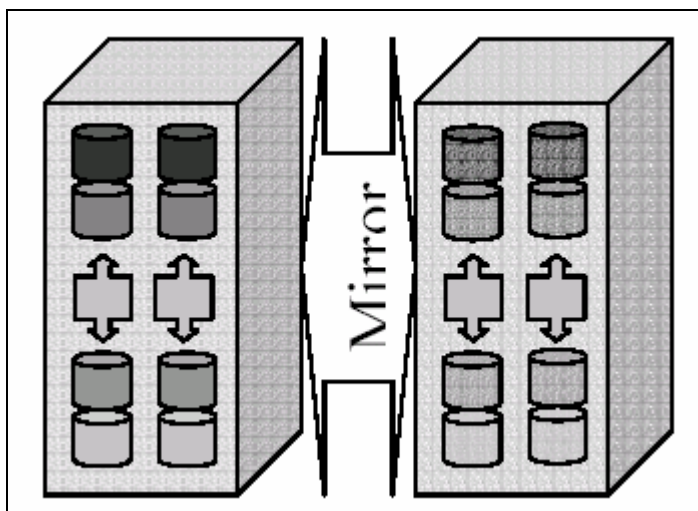


Figura 7. Quattro datapath in RAID 10

In questa figura, per creare quattro datapath in RAID 10 per SPD Server, l'array di dischi di sinistra è considerato il reale array di dati, mentre quello di destra è il mirror.

Per il primo datapath, i due dischi rappresentati in alto nell'array di sinistra sono combinati in un stripe-set e ciascuno è connesso ad un controller diverso, per evitare qualsiasi contesa. Il throughput combinato di questo stripe-set dovrebbe essere di circa 60MB/s. Nell'array di destra, i due dischi rappresentati in alto sono definiti come mirror dei rispettivi

dischi dell'array di sinistra. Questa configurazione fornisce un throughput molto simile a quello che si otterrebbe avendo quattro dischi connessi a quattro controller, soprattutto nelle operazioni di letture multiple, poiché il sottosistema di I/O può decidere di soddisfare la richiesta sia leggendo il dato dai dischi originali sia dai loro mirror. Configurando allo stesso modo anche gli altri tre dischi, si creano quattro datapath per l'I/O parallelo. Ciascun datapath è in stripe su due dischi, che sono in mirror nell'altro array. Il throughput totale ottenibile quando vengono lanciati quattro thread dovrebbe essere di circa 4*60MB/s, quindi 240MB/s per ciascun utente. Poiché lo striping e il mirroring così ottenuto è simmetrico per tutti i componenti, si ottiene anche un ragionevole bilanciamento del carico di lavoro in ambienti multi utente. Il limite teorico sarebbe di 640MB/s, in quanto i quattro controller possono funzionare a 160MB/s su due canali. Questi valori sono da considerarsi a titolo esemplificativo, in quanto diverse marche di dischi potrebbero fornire risultati diversi. Tuttavia, questi esempi possono essere considerati una buona linea guida.

3. Area indici

I file che compongono gli indici sono memorizzati nell'area indici. Per quanto riguarda la configurazione dei dischi che ospitano quest'area, essi dovrebbero essere configurati come gruppi di dischi in stripe-set (RAID 0) per ottenere le prestazioni di I/O migliori. Necessità di garantire la massima sicurezza e disponibilità di questo tipo di file, così come per la parte dati, potrebbero far decidere di adottare un sistema di ridondanza. In questo caso, un array di dischi in RAID 5, oppure una combinazione di mirroring e striping (RAID 10) potrebbero essere le scelte migliori.

4. Area temporanea (WORK AREA)

La work area è l'area nella quale vengono memorizzati tutti i file temporanei creati durante l'esecuzione dei processi paralleli che accedono al database, quali ad esempio file temporanei di utilità generati durante operazioni che richiedono dello spazio aggiuntivo, come la creazione di indici paralleli o l'operazione di ordinamento di file di grosse dimensioni.

Per ottenere prestazioni ottimali è sufficiente configurare quest'area su gruppi di dischi in stripe-set (utilizzando RAID0), sebbene anche in questo caso ragioni di sicurezza e massima disponibilità possono far decidere per la ridondanza (RAID5 o RAID 10), in considerazione del fatto che una perdita di disponibilità di quest'area provocherebbe l'inutilizzabilità di SPD Server.

In questo caso la scelta di utilizzare RAID10 è quella in grado di garantire il maggiore beneficio dal punto di vista prestazionale.

Uno sguardo al futuro.

Le potenzialità in termini di scalabilità e prestazioni fornite da SPD Server sono tali e talmente importanti nella gestione moderna dei dati aziendali che SAS ha deciso di inglobarne una parte nel suo modulo Base.

Nella versione 9 di SAS, infatti, è stata inserita una nuova engine di accesso ai dati chiamata SAS Scalable Performance Data Engine (SPD Engine), il cui scopo è quello di consentire il partizionamento dei data set sas in più file fisici, in modo da potervi accedere con una lettura parallela.

Ovviamente, le funzionalità della SPD Engine sono ben inferiori rispetto a quelle fornite da SPD Server, che oltre al semplice accesso parallelo ai dati costituisce un vero e proprio ambiente di database parallelo, che rispetta tutti i requisiti di sicurezza, affidabilità e controllo richiesti ai database.

La SPD Engine scatena, all'interno di un'unica sessione SAS, l'esecuzione di thread multipli, ciascuno responsabile della lettura di una partizione di data set, e ciascuno eseguibile parallelamente agli altri, laddove vi siano un numero sufficiente di processori, ottenendo così l'elaborazione immediata del dato. Questa configurazione consente una fattiva riduzione dei tempi di esecuzione delle operazioni di I/O, riducendo di conseguenza il tempo di esecuzione totale dei processi di ETL.

Approfondimenti

Per maggiori dettagli sugli aspetti prestazionali di SPD Server e della SPD Engine, si faccia riferimento alla Scalability Community sul sito del Supporto Tecnico:

<http://support.sas.com/rnd/scalability/>

Per informazioni generiche sul prodotto SPD Server:

<http://www.sas.com/technologies/dw/storage/spds/index.html>