



## Tips & Tricks

Il Supporto tecnico vi svela i segreti del mestiere

In questo numero: **La scalabilità di SAS®9**

# La scalabilità di SAS®9

## 1. Introduzione

La diffusione delle tecnologie web, ed in particolar modo del web browser quale principale strumento di accesso alle informazioni, ha provocato un innalzamento delle aspettative da parte degli utenti finali circa la rapidità con la quale qualsiasi tipo di applicazione fornisce la risposta alla loro query. Indipendentemente quindi dalla mole di dati interrogati o dalla complessità della richiesta, i moderni utenti si aspettano tempi di risposta equivalenti a quelli forniti da motori di ricerca tipo Google.

In aggiunta a questo aspetto, occorre considerare che la semplicità d'uso delle interfacce utente via web attira un numero sempre maggiore di utenti, accrescendo ulteriormente il carico dei sistemi informatici.

Per poter rispondere a queste accresciute esigenze degli utenti finali, SAS 9 è stato progettato in modo da sfruttare sia tutta la potenza elaborativa inutilizzata, sia il multithread, che consente di far eseguire porzioni di un unico processo a più processori all'interno della stessa macchina.

SAS 9 può realizzare sia una scalabilità verticale (usare una sola macchina con più processori e più canali di I/O) sia una scalabilità orizzontale (utilizzare una rete di macchine per disporre di più processori e più canali di I/O) per eseguire contemporaneamente più richieste da parte di più utenti oppure per eseguire parallelamente porzioni di una stessa richiesta fornendo i risultati nel minor tempo possibile.

## 2. Scalabilità solo dove serve

Non tutte le richieste possono essere eseguite contemporaneamente. Non ha senso, infatti, implementare il parallelismo per quelle operazioni che per loro natura devono essere eseguite sequenzialmente. Ecco perché non tutte le procedure SAS sono state riscritte in un'ottica di scalabilità.

In base alle esperienze dei clienti e alle analisi degli sviluppatori SAS, molte procedure SAS sono state identificate come particolarmente critiche per le prestazioni, soprattutto se utilizzate in elaborazioni lunghe e complesse. In SAS 9, alcune di queste procedure sono state modificate per trarre vantaggio da dispositivi di I/O più veloci e dai sistemi SMP, introducendo nelle classiche procedure SAS il supporto per il multithreading. Con la giusta configurazione hardware, queste procedure vi consentiranno di superare i colli di bottiglia presenti nelle vostre applicazioni più complesse.

Le procedure che sono state sinora modificate sono:

- In SAS BASE
  - PROC SORT\*
  - PROC SQL (group by, order by)
  - PROC SUMMARY/MEANS
  - PROC REPORT
  - PROC TABULATE
- In SAS/STAT
  - PROC REG\*
  - PROC GLM
  - PROC LOESS
  - PROC ROBUSTREG
- In Enterprise Miner
  - PROC DMREG\*
  - PROC DMINE\*

(\* Supporto delle letture parallele con SPD Engine)

## 3. Le basi della *Software Scalability*

Per *Software Scalability* si intende la capacità del software di sfruttare le caratteristiche dell'hardware che consentono di moltiplicare il numero di unità funzionali disponibili all'interno di un singolo spazio di processo. Queste unità funzionali includono sia le CPU sia i dispositivi di I/O (dischi e controller). La

capacità di avere più thread indipendenti in esecuzione su CPU separate che comunicano direttamente con la memoria centrale determina il livello di scalabilità di un software. La SAS Scalable Architecture (SSA) è appositamente progettata per fornire agli sviluppatori di applicazioni gli strumenti necessari per sfruttare al massimo le architetture SMP, che consentono a più thread di un processo di essere eseguiti in parallelo su diverse CPU condividendo nel contempo uno spazio di memoria comune.

La scalabilità dell'I/O comporta la combinazione di unità disco separate con RAID hardware o con il partizionamento software o con entrambi, per moltiplicare la velocità reale di trasferimento di un singolo file trasmettendo porzioni dello stesso su più unità disco e controller. Queste unità collegate fra loro agiscono come un disco unico per passare velocemente i dati alla memoria centrale. Il nuovo engine SAS SPDE consente di suddividere le operazioni di I/O su più file system e controller in modo da moltiplicare la velocità reale di trasferimento dei dati al codice dell'applicazione. Con questo nuovo engine, è anche possibile eseguire in parallelo su porzioni di dati separate alcune operazioni di estrazione quali le clausole WHERE. In aggiunta, l'engine SPDE supporta schemi sofisticati di indicizzazione a "bit" che incrementano ulteriormente la velocità delle query sui dati. Anche gli engine di SAS/ACCESS sono stati migliorati per poter sfruttare l'accesso parallelo laddove possibile, in modo da garantire che il trasferimento dei dati dalle strutture esterne sia il più veloce possibile.

#### **4. Il threading nei server e nell'accesso ai dati**

Un altro elemento della strategia di scalabilità di SAS 9 è costituito dal miglioramento delle prestazioni dei server SAS maggiormente utilizzati, in quanto essi forniscono le risposte alle numerose richieste degli utenti. Il SAS Olap Server e il SAS Metadata Server supportano molteplici connessioni utente simultanee e utilizzano il threading per elaborare efficientemente le query in parallelo, è quindi facile comprendere come la loro scalabilità sia un elemento fondamentale. In aggiunta alle procedure SAS e ai server, anche l'accesso ai dati è soggetto a scalabilità in quanto la velocità con la quale i dati vengono passati alle procedure e ai server deve essere il più possibile efficiente. Il prodotto SAS Scalable Data Server e il nuovo engine SPDE di SAS Base utilizzano il threading e il partizionamento dei dati per rendere più veloce l'accesso ai dati. Analogamente, l'introduzione del threading negli engine di SAS/ACCESS, ed in particolare in quelli per accedere a Oracle, DB2, Sybase e ODBC, consente di leggere i dati parallelamente anziché record per record. Ne deriva una drastica riduzione dei colli di bottiglia nelle operazioni di I/O.

#### **5. L'enorme potenziale delle tecniche di grid computing e di piping**

Il Grid Computing rappresenta un'altra soluzione alla sfida della scalabilità. Grazie alla funzionalità MP CONNECT di SAS/CONNECT, un utente può attivare più sessioni SAS per elaborare contemporaneamente diverse porzioni di un'esecuzione lunga e complessa, coordinandone l'esecuzione ed i risultati, utilizzando una singola macchina multiprocessore o una rete di macchine.

Il National Institute of Environmental Health Sciences (NIEHS) è riuscito a risolvere un problema estremamente impegnativo proprio impiegando questa tecnica. Utilizzando SAS/CONNECT in un ambiente grid, i ricercatori della NIEHS sono riusciti a ridurre del 99% il tempo necessario per analizzare più di 1,4 milioni di coppie di geni di cavie. Il processo, che eseguito su una singola macchina avrebbe impiegato circa 468 ore di elaborazione, è stato portato a compimento in poco più di 5 ore distribuendo l'elaborazione su un grid di 100 macchine.

SAS/CONNECT di SAS 9 offre inoltre una nuova funzionalità di piping che permette a un processo SAS che esegue una procedura di passare il proprio output come input di un'altra procedura. Oltre a migliorare il tempo di esecuzione, il vantaggio del piping consiste nella riduzione dello spazio su disco necessario per memorizzare i risultati intermedi.

#### **6. Il focus del Base di SAS 9**

Lo sviluppo di SAS Base si è focalizzato sulla riduzione del tempo reale di esecuzione delle operazioni principali, come l'ordinamento o la sommarizzazione di data set di ampie dimensioni. L'obiettivo della

*Software Scalability* consiste nel fornire risorse fisiche aggiuntive (CPU o canali di I/O) e avere un tempo reale di esecuzione ridotto proporzionalmente. Il focus in questo caso, quindi, è il tempo reale e non il tempo totale di CPU. Alcuni cicli di CPU aggiuntivi vengono infatti utilizzati per la gestione dei thread di processo sui diversi processori, ma mantenendo l'interazione tra i thread a un livello relativamente granulare, questo tempo aggiuntivo di CPU è trascurabile.

La porzione del programma originario che può essere eseguito in parallelo è quella che governa la percentuale di scalabilità della soluzione software. Per esempio, sebbene la PROC SORT possa leggere in parallelo porzioni di un data set partizionato, il data set di output deve essere scritto sequenzialmente per preservare l'ordinamento. Se la scrittura del data set di output impiega il 50% del tempo originale, allora soltanto il restante 50% del tempo può essere soggetto a riduzione. Questo valore rappresenta quindi il limite di scalabilità di questa operazione. È possibile ottenere ulteriori miglioramenti del tempo di esecuzione totale soltanto aumentando la velocità dei dispositivi di I/O che ospitano il data set di output.

La maggior parte delle procedure statistiche, come ad esempio la PROC REG, è prevalentemente vincolata alla CPU, per cui l'introduzione del multithread può tradursi in una perfetta scalabilità della risorsa CPU. Le procedure di SAS BASE quali SORT e SUMMARY, invece, possono facilmente essere vincolate alla CPU o all'I/O nelle diverse fasi della loro esecuzione. Per questo motivo, il raggiungimento della scalabilità di queste procedure avviene nel momento in cui il tempo di esecuzione di tutto il task corrisponde alla somma del tempo di I/O necessario per la lettura dei dati di input e per la scrittura dell'output. Naturalmente, la velocità dell'I/O e delle CPU deve essere bilanciata, poiché non si possono ottenere prestazioni ottimali se si utilizza una macchina multiprocessore con un sistema di I/O lento. Durante lo sviluppo di SAS 9, un dispositivo di input è stato ritenuto ad alte prestazioni soltanto se la sua velocità di trasferimento era di almeno 50 mbyte/sec.

## 7. L'I/O per i file di utilità

Oltre a valutare come utilizzare più CPU contemporaneamente per stare al passo con la velocità di lettura da disco, i nostri sviluppatori hanno anche valutato attentamente come le nuove procedure multithread interagiscono con i file di utilità. Molti algoritmi di calcolo sono stati modificati per ridurre la frequenza di accesso al disco. Per esempio, la nuova procedura di ordinamento utilizza un modello di lettura predittivo che consente la ri-lettura dei file di utilità dell'ordinamento ad una velocità quasi pari a quella di una lettura sequenziale. In aggiunta, le procedure multithread di SAS Base possono contare sulla possibilità di definire percorsi multipli per la memorizzazione dei file di utilità, permettendo quindi di ridurre le contese di lettura e scrittura su un singolo dispositivo. Ad esempio, se un file di utilità deve essere ordinato, il servizio di ordinamento può ora accedere a uno spazio su disco residente su un file system diverso da quello di input. In tal modo, la lettura del file di utilità che deve essere ordinato non entra in conflitto con la scrittura di file di ordinamento temporanei e, allo stesso modo, la lettura dei file di utilità dell'ordinamento non contenderà la risorsa disco per la ri-scrittura del file di utilità di output. Sia la PROC SUMMARY che la PROC SQL sono state riprogettate in base a questo nuovo schema.

L'opzione, disponibile con SAS 9, che consente di impostare questa caratteristica dei file di utilità va definita all'avvio di SAS e si chiama **UTILLOC**. Essa consente all'utente di indicare uno o più percorsi per la memorizzazione dei file di utilità da parte delle procedure multithread. Questa impostazione estende e non sostituisce la SASWORK. Specificare più percorsi nell'opzione UTILLOC potrebbe potenzialmente migliorare il throughput delle nuove procedure multithread che sono state progettate per utilizzare questa funzionalità, a patto che i percorsi si trovino in dispositivi di I/O separati. Se l'opzione non viene impostata all'avvio di SAS, assumerà automaticamente lo stesso percorso della SASWORK.

Per sfruttare al meglio questa funzionalità, le procedure SUMMARY, TABULATE e REPORT attualmente utilizzano il primo percorso indicato nell'opzione UTILLOC quale spazio di utilità tradizionale, mentre le operazioni di ordinamento interno che esse scatenano utilizzano il secondo percorso quale spazio su disco per la memorizzazione dei file di utilità necessari. Altre procedure, che pure fanno uso di routine di ordinamento interno, quali ad esempio la procedura SQL, continuano a usare la SASWORK per i loro file di utilità tradizionali, ma sfruttano anch'esse il secondo percorso indicato nell'opzione UTILLOC per l'ordinamento interno. Alla luce di questa particolarità, nel caso in cui si abbiano a disposizione solo due file

system per i file temporanei, consigliamo di indicare quale primo file system dell'opzione UTILLOC quello della SASWORK, e come secondo quello dell'altro file system, come indicato nell'esempio seguente:

**-work "d:" -utilloc "(d: 'e:)"**

Per l'esatta sintassi dell'opzione relativamente al proprio sistema operativo, si consiglia di fare riferimento alla guida "SAS Companion" specifica.

## 8. Nuove opzioni per la gestione dei thread

Per consentire il controllo e il tuning delle procedure multithread, sono state create due nuove opzioni SAS globali: CPUCOUNT e THREADS. Queste due semplici opzioni possono essere usate per influenzare, se non necessariamente controllare, il comportamento delle nuove procedure multithread. Ogni procedura, infatti, è libera di interpretare l'impostazione di queste opzioni nella maniera più appropriata. L'eccezione a questa regola è l'impostazione negativa dell'opzione THREADS, cioè NOTTHREADS. In questo caso, in assenza di diversa impostazione per il singolo processo, tutte le procedure non faranno uso del multithread.

L'opzione THREADS è un semplice "interruttore" per attivare o disattivare il threading di SAS 9, che per default è sempre attivo. In realtà, le procedure che sono in grado di sfruttare il threading interpretano tale opzione come una possibilità di usare il threading, che non significa che esse lo useranno sicuramente. Le procedure, infatti, usano degli algoritmi propri per determinare se l'uso del multithread possa o meno apportare benefici nell'esecuzione di un dato compito. Per esempio, la procedura SUMMARY non userà i thread se viene impostata un'istruzione di BY, in quanto si potrebbe assistere a un potenziale degrado delle prestazioni nel caso in cui fosse presente un numero consistente di gruppi di BY molto piccoli. Tuttavia, la procedura SUMMARY è dotata di una nuova opzione THREADS a livello di istruzione PROC che consente di sostituire un'eventuale opzione NOTTHREADS a livello globale e di forzare l'uso dei thread anche in presenza dell'istruzione BY.

L'opzione CPUCOUNT fornisce indicazioni alle procedure su come impostare il loro modello di thread. Essa non limita il numero di thread creati dall'applicazione, né il numero di CPU che potrebbero essere usate dalla sessione SAS. Per default, CPUCOUNT è impostata al valore ACTUAL, che significa che SAS determinerà quante CPU sono disponibili per il suo processo e imposterà l'opzione CPUCOUNT in base a tale valore. Il valore ACTUAL potrebbe essere diverso dal numero di CPU fisicamente installate sulla macchina se al processo SAS è stata imposta una restrizione da parte del sistema operativo. CPUCOUNT può essere impostata artificialmente più alta o più bassa per alterare il comportamento delle procedure ma deve essere reimpostata ad ACTUAL se si vuole che SAS interroghi nuovamente il sistema operativo per determinare quante CPU sono effettivamente disponibili. Se CPUCOUNT viene impostata ad 1, l'algoritmo della procedura multithread potrebbe decidere di non utilizzare thread ausiliari anche se l'opzione THREADS è attiva.

Il numero reale di thread attivati da una specifica procedura dipende da diversi fattori, tra i quali il numero di CPU. Se il CPUCOUNT fosse impostato a un valore inferiore rispetto al numero reale di CPU disponibili e anche inferiore al numero minimo di porzioni di processo eseguibili in parallelo previste dal modello applicativo, il processo potrebbe comunque essere eseguito usando un numero di CPU maggiore rispetto a quello impostato nell'opzione. Per poter veramente limitare il numero di CPU che una singola sessione SAS può utilizzare, occorrerà quindi usare un tool di sistema, come ad esempio il comando *prset* di SUN Solaris®. Analogamente, se l'opzione CPUCOUNT fosse impostata a un valore superiore rispetto al numero di CPU effettivamente disponibili potrebbe verificarsi un degrado generale delle prestazioni di SAS.

## 9. Il nuovo ordinamento threaded di SAS 9

Base di SAS 9 comprende una nuova utility portabile di ordinamento multithread ad alte prestazioni. Questa utility viene usata dalla procedura SORT quando l'opzione THREADS è attiva e l'ordinamento SAS viene

selezionato da logiche di sistema o esplicitamente impostando l'opzione SORTPGM=SAS. Questo nuovo ordinamento utilizza nuovi algoritmi di calcolo che consentono di ottenere i risultati in un tempo fino ad 8 volte inferiore rispetto al tempo dell'ordinamento tradizionale, a patto che vi sia sufficiente spazio di I/O. Data una risorsa di I/O, quanto maggiore è la dimensione della chiave di ordinamento rispetto alla dimensione del record, tanto maggiore sarà il vantaggio che si potrà trarre dall'ordinamento parallelo.

L'ordinamento threaded crea una rete di processi di ordinamento che contengono sia gli elementi di peer che di pipeline, all'interno della quale gli ordinamenti in memoria e l'I/O si sovrappongono efficacemente. In generale, l'ordinamento si divide tra elementi interni (residente in memoria) ed esterni (basati su file di utilità). Per accelerare l'ordinamento interno, thread di ordinamento multipli accettano blocchi di input da una o più partizioni di dati in parallelo. Vengono creati dei blocchi compatti di chiavi per ridurre al minimo gli accessi alla cache della CPU durante l'ordinamento dei blocchi. I blocchi ordinati vengono poi elaborati da una fase di merging che utilizza thread multipli per evitare la latenza dell'accesso alla memoria centrale nel momento in cui si raccolgono i record dai vasti spazi di memoria che sono presenti nei moderni sistemi a 64bit.

Il grafico sottostante illustra il potenziale vantaggio delle prestazioni della nuova procedura di SORT. Il grafico riporta il tempo reale di esecuzione di un ordinamento su data set di diverse dimensioni utilizzando il SORT della versione SAS 8.2 e il nuovo SORT di SAS 9. La chiave di ordinamento utilizzata è un numero di 8 byte ed il record ha una lunghezza di 64 byte. Generalmente, poiché le operazioni sulla chiave sono quelle che consumano la maggior parte del tempo di CPU necessario all'ordinamento del file, più la chiave è ampia, maggiore sarà il beneficio sulle prestazioni del nuovo SORT threaded.

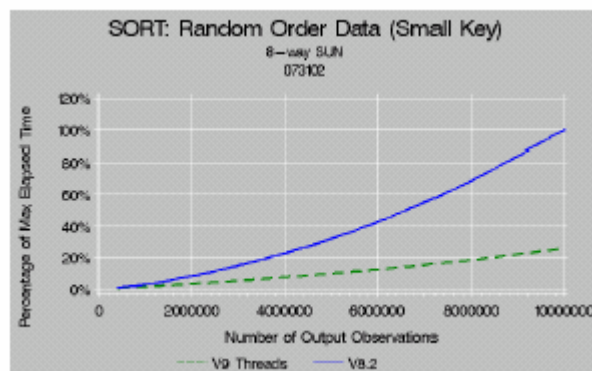


Fig. 1 Nuovo ordinamento threaded interno

Nell'ambito dell'ordinamento esterno, molti miglioramenti sono stati introdotti per rendere più efficiente l'interazione con i dispositivi di I/O che ospitano i file di utilità. Innanzi tutto, per ridurre lo spazio su disco necessario per memorizzare questi file, e di conseguenza il numero totale di byte che devono essere letti e riscritti, l'ordinamento threaded non memorizza più le chiavi di ordinamento all'interno dei file di utilità. Se necessario, queste chiavi vengono rigenerate in parallelo durante la fase di merging. In tal modo, lo spazio occupato dai file di utilità temporanei non è mai più grande del file di input, mentre nell'ordinamento tradizionale tale spazio poteva essere anche il doppio.

Molto più importante della dimensione totale del file, tuttavia, è il metodo di accesso al disco. Un buon accesso casuale al file può ridurre drasticamente la velocità di lettura per il file di utilità dell'ordinamento. Il nuovo ordinamento threaded utilizza una nuova tecnologia, chiamata *disk head seek reduction*, che consente di accedere per la fase di merging ad un singolo file di utilità contenente dati relativi a più esecuzioni dell'ordinamento alla stessa velocità che si avrebbe se il file fosse completamente ordinato. Un thread dedicato esegue contemporaneamente sullo stesso file di utilità le letture predittive e le costruzioni delle chiavi in modo da garantire che non esista alcuna differenza tra il tempo necessario alla fase di merging e una semplice operazione di copia del file.

## 10. Nuove opzioni per il controllo del SORT

Come per altre procedure threaded, anche nella procedura SORT è possibile impostare l'opzione THREADS/NOTTHREADS per cambiare localmente l'impostazione globale. Per verificare che si stia effettivamente usando l'ordinamento threaded portabile di SAS, occorre impostare l'opzione SAS MSGLEVEL uguale a 'I'. In aggiunta, vi è una nuova opzione SAS globale, SORTEQUAL/NOSORTEQUAL che consente di disattivare a livello globale la logica dell'ordinamento in base alla quale i record con chiave uguale vengono scritti nel file di output nello stesso ordine con cui si trovavano nel file di input. Questo comportamento, raramente necessario, costa sia in termini di CPU che di consumo di memoria. Per ragioni di compatibilità con le precedenti versioni, il valore di default è SORTEQUAL, ma si consiglia di impostare tale opzione come NOSORTEQUAL.

Il consumo di memoria dell'ordinamento può ancora essere controllato dall'opzione SORTSIZE, come avveniva nelle precedenti versioni, ma su molte piattaforme il valore di default è impostato a MAX. In questo caso, l'opzione globale REALMEMSIZE controllerà il valore massimo di memoria utilizzata per l'ordinamento interno. Questa modifica è coerente con l'intenzione di consolidare il controllo della memoria nelle opzioni MEMSIZE e REALMEMSIZE escludendolo invece dalle opzioni specifiche delle singole procedure, come SORTSIZE e SUMSIZE. L'opzione REALMEMSIZE indica la quantità di memoria reale (non virtuale) che si presume sia disponibile per il processo SAS. Su alcune piattaforme, il sistema operativo può determinare questo valore dall'hardware. Su altre, invece, non può essere determinato in questo modo e deve quindi essere impostato dall'utente.

Due nuove opzioni sono state inserite nella procedura SORT di SAS 9: DUPOUT e OVERWRITE. La prima consente di specificare il nome di un data set che conterrà le osservazioni scartate dalle opzioni di eliminazione dei duplicati (NODUPREC e NODUPKEY). L'opzione OVERWRITE consentirà alla procedura SORT di eliminare il data set di input prima della scrittura del data set di output ordinato. Questa opzione ridurrà del 50% lo spazio necessario all'esecuzione del SORT. Gli ordinamenti interni non avranno bisogno di spazio aggiuntivo, mentre quelli esterni avranno bisogno, nella SASWORK, soltanto dello spazio sufficiente per eseguire una copia del data set di input. Nessuno spazio aggiuntivo sarà necessario nella directory di partenza. Lo svantaggio di questa opzione è rappresentato dal rischio di perdita del data set di input in caso di problemi durante l'esecuzione della fase finale di scrittura. Tuttavia, dato che il data set ordinato di output viene scritto praticamente nello stesso spazio su disco del data set di input, è meno probabile che possano verificarsi problemi come errori di scrittura o di disco pieno.

## **12. Procedure per il reporting in versione multithread**

Le principali procedure per il reporting di SAS Base, quali SUMMARY/MEANS, TABULATE e REPORT sono state arricchite di funzionalità multithread con SAS 9. Tutte queste procedure condividono un servizio di sommarizzazione comune che gestisce la classificazione dei dati e la generazione delle statistiche. Poiché queste procedure hanno diverse modalità di operare, quali ad esempio con o senza istruzione CLASS, con o senza BY, con o senza variabili di analisi, il servizio di sommarizzazione sottostante usa modelli di threading distinti, ognuno con le proprie caratteristiche.

L'approccio più semplice per trasformare un processo di sommarizzazione in multithread consisterebbe nel suddividere i dati in N parti separate, sommare ciascuna parte in parallelo, e quindi unire i risultati. Sfortunatamente, poiché il processo di classificazione consuma memoria per ciascuna combinazione univoca dei valori contenuti nella variabile di classe, separare il processo in N spazi potrebbe moltiplicare per N la quantità di memoria necessaria. Per evitare questo inconveniente, il servizio di sommarizzazione suddivide il processo in una serie di fasi che sono collegate tra loro. I risultati intermedi di una fase vengono passati alla successiva insieme a un blocco di record. In questo modo, più CPU possono essere utilizzate nel processo di sommarizzazione senza aumentare la quantità di memoria necessaria. In aggiunta, questo modello elimina quasi completamente la contesa di accesso alla memoria condivisa da parte di più thread. Lo svantaggio di questo modello è rappresentato da un potenziale sbilanciamento di carico delle varie fasi, in quanto ciascuna esegue operazioni diverse. Per completare una fase potrebbe essere necessaria una maggiore quantità di tempo della CPU rispetto alla fase seguente, generando così dei colli di bottiglia e una mediocre scalabilità. Per risolvere questo problema viene utilizzata la tecnica del peer threading, ripristinando il bilanciamento del

modello. Idealmente, ogni unità di lavoro assegnata ad un thread del modello sarebbe approssimativamente uniforme e non si formerebbero colli di bottiglia.

Senza variabili di CLASS (o di GROUP), il modello di threading per il servizio di sommarizzazione è relativamente semplice. Un numero N di peer thread aggregano le statistiche in N spazi separati e questi spazi vengono poi uniti alla fine dell'elaborazione di tutti i dati. N, in questo caso, è solitamente uguale al valore di CPUCOUNT. Il modello si adatta facilmente all'aumento del carico e ha un tempo di esecuzione reale uguale al tempo di lettura dei dati. Di conseguenza, ad esempio, i seguenti due passi dovrebbero avere approssimativamente lo stesso tempo di esecuzione reale se fosse disponibile un numero di CPU sufficiente da eguagliare la velocità delle operazioni di I/O:

```
data _null_; set foo;  
run;  
proc means data=foo; output out=bar;  
run;
```

Il grafico sottostante illustra la scalabilità della procedura SUMMARY senza istruzione CLASS. In una situazione idealmente perfetta, avremmo la linea tratteggiata della SUMMARY threaded direttamente sopra la linea punteggiata che rappresenta l'I/O, indicando che il tempo effettivo per sommare i dati è uguale al tempo necessario per leggere i dati da disco.

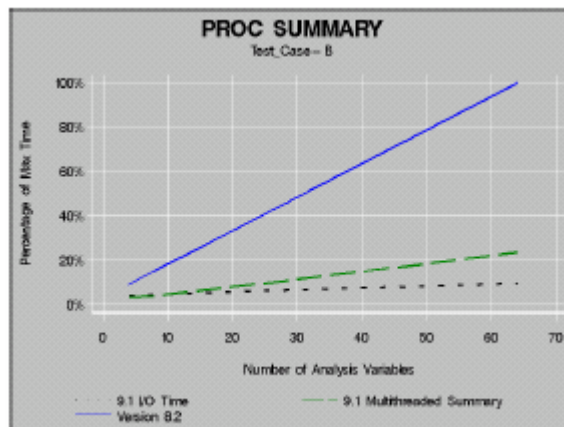


Fig. 2: procedura Summary senza CLASS, eseguita con 8 CPU

Poiché la classificazione è sostanzialmente un processo di ordinamento, l'utilizzo di un'istruzione CLASS con una di queste procedure, richiede l'impostazione di un modello di parallelismo più complesso e maggiormente soggetto a sbilanciamenti. Un tipico parallelismo della classificazione che costruisce il livello NWAY è costituito da quattro fasi: una fase per verificare i valori mancanti della variabile di classificazione, una fase multithread che normalizza i valori della variabile di classe, una fase per ricercare lo spazio di aggregazione da una tabella in memoria e una fase multithread finale per aggiornare le statistiche in ciascun spazio di aggregazione. Queste fasi parallele ricevono i dati da uno o più thread di I/O dedicati.

Il grafico sottostante rappresenta il confronto tra SAS V8 e SAS 9 eseguiti su una macchina a 8 CPU. La SUMMARY eseguita ha un numero costante di variabili di analisi (4) mentre il numero di variabili di classe varia tra 2 e 64. La cardinalità dei dati è approssimativamente di 100/1, e cioè un'osservazione di output nel livello di NWAY ogni cento in input.

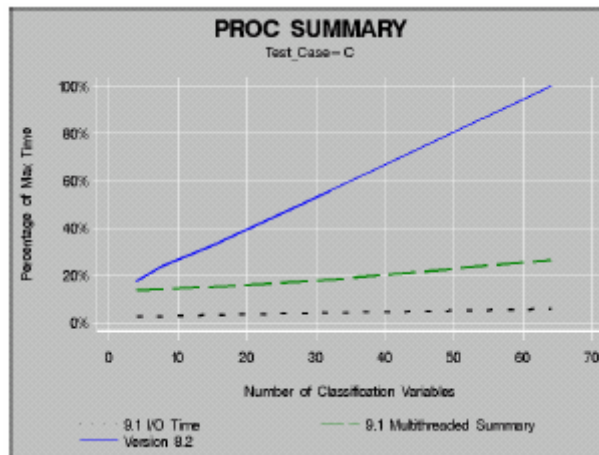


Fig. 3: Procedura SUMMARY con CLASS, eseguita con 8 CPU

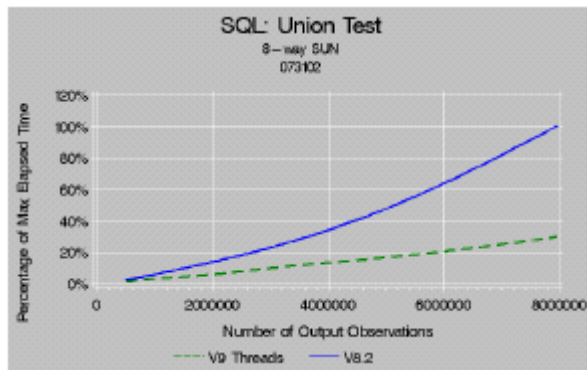
Come potete osservare, la scalabilità aumenta all'aumentare del numero di variabili di classificazione, ma esiste ancora una differenza significativa tra il tempo di esecuzione della procedura e il tempo di I/O, dovuta a una sovrapposizione dei thread e a sbilanciamenti nel parallelismo. I futuri sviluppi del software saranno focalizzati a ridurre la differenza tra le due linee, in modo da ottenere un processo di classificazione che stia al passo con la velocità dell'I/O.

### 13. Elaborazione di sommarizzazioni esterne

Quando si esaurisce la quantità di memoria a disposizione per il processo di espansione di una tabella degli incroci, i risultati parziali vengono dirottati in un file di utilità per liberare la memoria. Nelle versioni precedenti, il processo utilizzava una tecnica di unioni ripetute che manteneva su disco soltanto i valori univoci delle variabili di classe. Questa tecnica consente di ridurre al minimo le dimensioni dei file di utilità ma quando la cardinalità dei dati aumenta e la richiesta di spazio disco eccede ampiamente la memoria, essa diventava inefficiente. In SAS 9, la sommarizzazione utilizza ora una tecnica di unione a più livelli per generare gli incroci di NWAY su disco. Questa soluzione rappresenta un valido compromesso tra velocità e occupazione di spazio su disco, in quanto il file di utilità può ora contenere valori duplicati della variabile di classe anche prima della fase di unione finale.

### 14. SQL threaded

In passato, la procedura SQL ha fatto uso delle routine portabili di ordinamento per eseguire le operazioni di ordinamento necessarie all'implementazione delle clausole *order by*, *group by* e altre operazioni come i *merge join*. In SAS 9, la procedura SQL utilizza la nuova funzionalità di ordinamento multithread per ordinare le righe. Poiché il tempo di esecuzione dell'ordinamento può influenzare la maggior parte delle query SQL, l'introduzione della tecnologia dell'ordinamento parallelo può ridurre il tempo di esecuzione delle query così come è avvenuto per i tempi della PROC SORT. Questa funzionalità è automatica se l'opzione THREADS è attiva. Il grafico sottostante illustra l'impatto dell'ordinamento threaded sull'operazione SQL di *union*, una delle numerose operazioni che richiedono un ordinamento delle righe.



## 15. Conclusioni

La nuova Intelligence Platform di SAS 9 consente quindi di far fronte alle esigenze sempre più pressanti di scalabilità e di crescita che le aziende si trovano ad affrontare.

La capacità di essere scalabili è ora resa possibile dalle threaded PROCs (SAS/Base, SAS/Stat, SAS Enterprise Miner), dai SAS Foundation Servers threaded e multi utente (SAS OLAP Server, SAS Metadata Server), da SAS Scalable Performance Data Server, dalla nuova engine SPD presente in SAS/Base, dal supporto all'indirizzamento a 64-bit della memoria.

Alla scalabilità sui sistemi SMP, la nuova Intelligence Platform aggiunge a SAS 9 la capacità di sfruttare le tecnologie di Grid Computing, aprendo alle aziende nuovi scenari architetturali in grado di sfruttare al meglio le risorse hardware/software già presenti in azienda.

Per il raggiungimento della migliore scalabilità possibile è tuttavia necessario affrontare la problematica da una visione di insieme, affiancando alle caratteristiche tecnologiche del software un corretto disegno dello scenario hardware coinvolto. Il raggiungimento di uno scenario altamente scalabile e performante, in grado di far fronte alle aspettative degli utenti, è infatti possibile solo integrando i diversi elementi coinvolti :

- la scalabilità del software
- la scalabilità dell'hardware

Con SAS 9 la scalabilità è garantita oltre che dalle caratteristiche multi-threads delle sue componenti e dal processo di re-engineering a cui sono state sottoposte alcune sue parti, anche dalla possibilità di agire in modo puntuale su parti del codice generato con opzioni specifiche pensate per ottimizzare l'utilizzo delle risorse (CPU, Memoria, throughput sull'I/O, occupazione degli spazi disco).

A queste caratteristiche occorre affiancare ambienti scalabili, utilizzando servers SMP o reti di servers, e disegnando il sistema di I/O in modo opportuno affinché le potenzialità "parallele" del software possano essere sfruttate.

Tutto ciò costituisce un lavoro di analisi e verifica dell'intera architettura hardware e software e delle specifiche esigenze applicative. Nel team di progetto, quindi, sarà di fondamentale importanza il coinvolgimento e l'apporto degli esperti di sistema e architetture, che devono affiancare gli applicativi sin dalle prime fasi di progetto.