

## OPTIMISATION DES CUBES OLAP, DE LA CRÉATION À LA NAVIGATION

Vous travaillez avec des cubes OLAP. Vous avez été confronté, ou le serez peut-être, à des problèmes tels que : des manques de mémoire, des manques d'espace disque ou des temps de réponse dégradés (suite par exemple à une augmentation de la volumétrie).

Le but de cet article est de vous présenter les différentes options SAS et actions possibles à réaliser afin d'optimiser la création d'un cube OLAP et la navigation dans ce type de structure.



### Caractéristiques :

Catégories : SAS® OLAP Server

OS : Windows, Unix, z/OS

Version : SAS® 9.3

Vérifié en Septembre 2012

### Table des matières

Quelques rappels sur un cube OLAP .....	1
Structure d'un cube OLAP .....	1
Différenciation entre la création d'un cube OLAP et la navigation dans un cube OLAP .....	2
Optimisation de la création d'un cube OLAP .....	3
Présentation des deux phases incluses lors de la création .....	3
Options SAS disponibles pour optimiser la création d'un cube OLAP.....	3
Options de traces lors de la création d'un cube OLAP.....	5
Optimisation de la navigation dans un cube OLAP .....	6
Prendre en compte l'environnement depuis lequel le cube OLAP est accédé.....	6
Activer la log ARM et construire de nouvelles agrégations .....	6
Options de traces pour le serveur OLAP .....	9
Options d'optimisation pour le serveur OLAP .....	9
Conclusion .....	10
Liens utiles .....	11

### Quelques rappels sur un cube OLAP

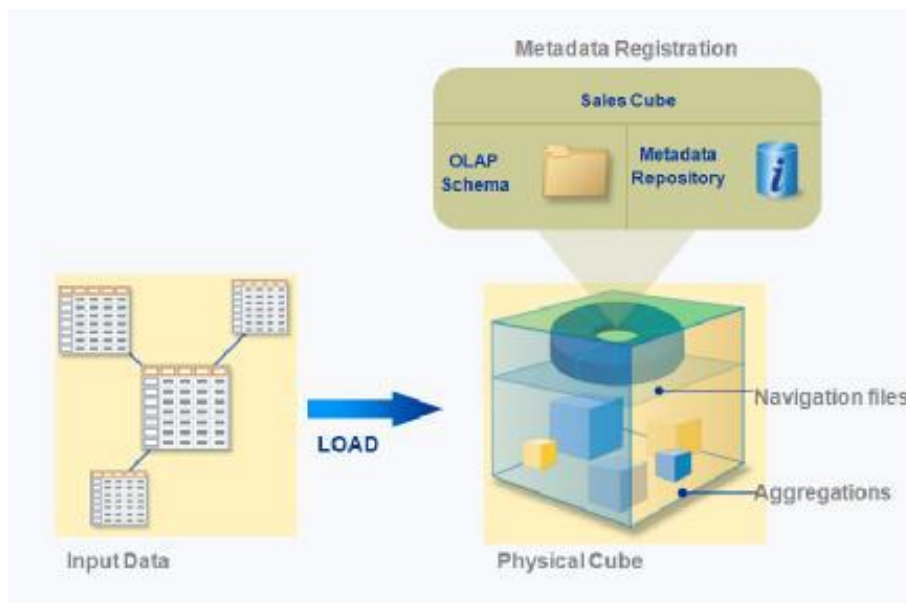
#### Structure d'un cube OLAP

Avant d'expliquer comment optimiser la création et l'interrogation d'un cube OLAP, voici quelques rappels sur la structure d'un cube OLAP.

Il est important de comprendre qu'un cube OLAP se décompose en 2 grandes parties :

- La structure physique : elle comprend tous les fichiers créés sur le système d'exploitation, incluant les fichiers associés aux agrégations (fichiers SPDE -- Scalable Performance Data Engine --, indexes, etc.), le fichier cube et autres fichiers spécifiques.
- La définition du cube dans les métadonnées : elle comprend le nom des dimensions, des hiérarchies, des niveaux et mesures, la définition des mesures calculées, ainsi que l'ordre de navigation et est stockée dans un schéma OLAP, défini au niveau du référentiel de métadonnées.

Une structure OLAP peut ainsi se synthétiser comme ci-dessous :



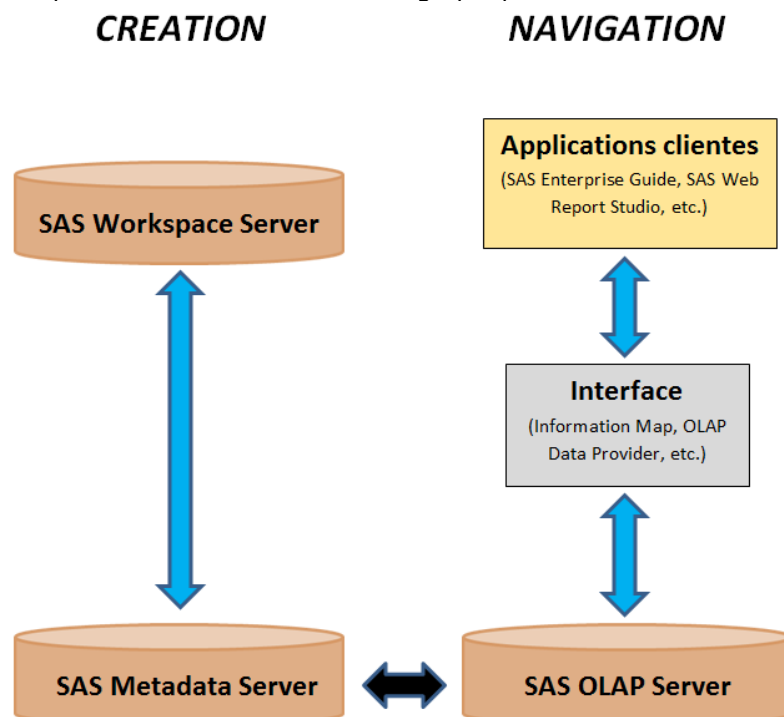
### Différenciation entre la création d'un cube OLAP et la navigation dans un cube OLAP

Lors des phases de création et d'interrogation d'un cube OLAP, différents serveurs SAS sont nécessaires. Ainsi, il faut absolument différencier les deux phases :

**La phase de création** d'un cube OLAP fait appel au SAS® Metadata Server pour y référencer la structure au niveau du schéma OLAP mais surtout utilise le SAS Workspace Server pour exécuter la procédure OLAP. A ce stade, le serveur OLAP ne sera quant à lui utilisé que pour valider des expressions MDX associées à des mesures ou membres calculés.

**La phase de navigation** dans un cube OLAP s'appuie également sur le SAS Metadata Server mais surtout sur le SAS® OLAP Server afin de pouvoir exécuter les différentes requêtes MDX.

Ces phases sont illustrées dans le graphique suivant :



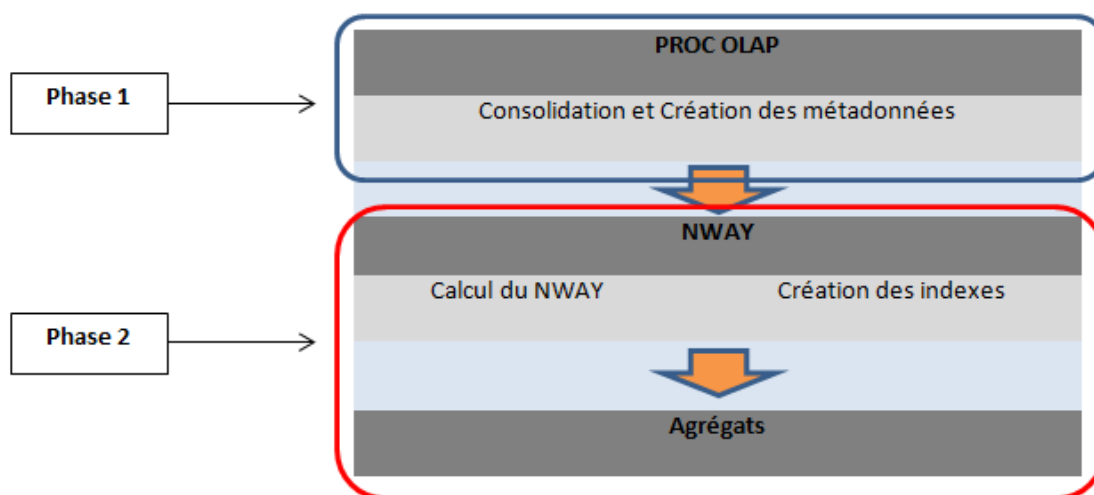
Sachant cela, toute option positionnée au niveau du serveur OLAP n'aura donc aucune incidence sur la création d'un cube OLAP.

## Optimisation de la création d'un cube OLAP

### Présentation des deux phases incluses lors de la création

Lors de la création d'un cube OLAP, deux phases sont à distinguer, correspondant au fonctionnement de la procédure OLAP.

Le schéma ci-dessous reprend brièvement ces deux phases :



La première phase comprend la préparation interne aux métadonnées qui inclut la consolidation des métadonnées et le tri des membres, ainsi que la création des métadonnées internes au cube.

Les métadonnées du cube comprennent :

- l'emplacement des données
- la structure du cube (dimensions, hiérarchies, niveaux, etc.)
- les membres OLAP et les relations entre eux (définition de l'ordre de navigation), les formats, les propriétés de membre
- les membres calculés (définition et expression)

Il est important de noter que pour les mesures calculées, seule leur définition (incluant l'expression de calcul) est enregistrée dans les métadonnées. Cela signifie que ces mesures seront toujours calculées à la volée et impacteront les temps de réponse.

La deuxième phase, quant à elle, comprend la création du NWAY et des tables d'agrégations, ainsi que leurs indexes associés. Pour rappel, le NWAY correspond à l'agrégation la plus large, c'est-à-dire celle incluant tous les croisements des niveaux les plus fins à partir desquels tout autre croisement peut être calculé à la volée. Les indexes permettent un accès plus rapide et plus efficace aux données. Il est possible de ne pas les construire mais il en résulterait des performances moins bonnes lors de l'utilisation du cube. Ce n'est donc pas conseillé bien que techniquement possible.

### Options SAS disponibles pour optimiser la création d'un cube OLAP

Le tableau suivant récapitule les différentes options SAS disponibles, selon que leur type se situe au niveau système ou de la procédure OLAP, et leur périmètre applicable à la phase 1 ou 2 :

<b>Option</b>	<b>Type</b>	<b>Description</b>	<b>Périmètre</b>
<b>MEMSIZE</b>	SYSTEME	Spécifie la limite de mémoire virtuelle qui peut être utilisée pour une session SAS	Phases 1 et 2

<b>REALMEMSIZE</b>	SYSTEME	Spécifie la mémoire physique que SAS peut espérer allouer	Phases 1 et 2
<b>SUMSIZE</b>	SYSTEME	Spécifie la limite en termes d'allocation mémoire pour la consolidation	Phases 1 et 2
<b>SORTSIZE</b>	SYSTEME	Spécifie la limite en termes d'allocation mémoire pour le tri	Phases 1 et 2
<b>UTILLOCn</b>	SYSTEME	Spécifie un ou plusieurs emplacements du système de fichiers où SAS peut stocker les fichiers utility	Phases 1 et 2
<b>SPDEUTLILOC</b>	SYSTEME	Spécifie un ou plusieurs emplacements du système de fichiers où SAS peut écrire les fichiers utility temporaires	Phase 2
<b>INDEXSORTSIZE</b>	PROC OLAP	Indique la mémoire maximum disponible par thread pour la création du NWAY et des indexes	Phase 2
<b>MAXTHREADS</b>	PROC OLAP	Spécifie le nombre maximum de threads utilisés pour créer les indexes de manière asynchrone	Phase 2
<b>ASYNINDEXLIMIT</b>	PROC OLAP	Permet de contrôler le nombre d'indexes créés à la fois pour une table d'agrégations	Phase 2
<b>ADDNWAYINDEX</b>	PROC OLAP	Permet de créer les indexes du NWAY après la création du cube via la proc OLAP, où l'option NOINDEX avait été positionnée pour le NWAY	Phase 2
<b>INDEX   NOINDEX</b>	PROC OLAP	Permet d'indiquer la création ou la non création des indexes pour les tables d'agrégations	Phase 2
<b>CONCURRENT</b>	PROC OLAP	Spécifie le nombre maximum d'agrégations à créer en parallèle.	Phase 2
<b>COMPRESS</b>	PROC OLAP	Permet de compresser les tables d'agrégations (y compris le NWAY), ce qui réduit la taille du cube	Phase 2

Quelques recommandations sont nécessaires au regard de ces options.

La valeur de MEMSIZE peut être positionnée à 0 (dans ce cas, toute la mémoire disponible est allouée à la session SAS), mais cela est vivement déconseillé en environnement de production.

L'option FULLSTIMER, en combinaison avec l'option MEMSIZE=MAX, doit être privilégiée afin de pouvoir déterminer la mémoire requise pour construire le cube.

La règle ci-dessous doit être utilisée pour positionner les valeurs de MEMSIZE, REALMEMSIZE et SUMSIZE :

REALMEMSIZE=SUMSIZE=80% (MEMSIZE)

Il est conseillé de spécifier un emplacement pour les fichiers temporaires situé sur un autre disque que celui utilisé pour le stockage des données source. Cela permet ainsi d'éviter tout risque de contention de disque.

Si des erreurs de mémoire ('Out of memory' par exemple) apparaissent à la création du cube, nous vous encourageons à augmenter la valeur de l'option SUMSIZE (et/ou la valeur des options MEMSIZE et REALMEMSIZE).

Si par contre il s'agit d'erreurs liées à la création d'index ('Indexes cannot be created for the NWAY aggregation' par exemple), les options INDEXSORTSIZE, MAXTHREADS et CONCURRENT devront être ajustées.

Si l'ajustement de ces options ne permet toujours pas de construire le cube, il est alors possible d'augmenter la valeur de MEMSIZE et de positionner l'option ASYNCINDEXLIMIT (valeur à 2 par exemple).

Enfin, si les erreurs sont relatives à la création du fichier utility ('ERROR: Utility file write failed' par exemple), la première chose à vérifier est de s'assurer que l'emplacement positionné pour les fichiers temporaires dispose de suffisamment d'espace disque. Si un seul emplacement est spécifié, une solution peut être d'utiliser les options UTILLOC et SPDEUTILLOC afin de positionner plusieurs emplacements (toujours sur un disque différent de celui où sont stockées les données source, si possible).

#### Options de traces lors de la création d'un cube OLAP

Plusieurs options sont disponibles à travers SAS afin d'ajouter des traces complémentaires dans la log du traitement pour disposer de plus d'informations sur les valeurs éventuelles à positionner pour certaines options.

L'option **SYSSUMTRACE** permet d'ajouter des traces concernant la consommation mémoire durant la phase 1 de création d'un cube OLAP.

Elle se positionne avec la syntaxe suivante :

```
%let syssumtrace=3;
```

Les informations affichées dans la log sont de ce type :

IN PROGRESS: WRITE NWAY TO DISK  
(TSTLVL TIMING)Begin time for writing nway to disk: 14H:45M:1S  
*NOTE: The initial memory limit for classification tables was 1219133K bytes. Actual memory acquired was 2368K bytes.*  
*NOTE: The utility file buffer size selected was 16K bytes.*

Afin de connaître les différentes valeurs positionnées pour les options de mémoire et de performance, la procédure OPTIONS peut être exécutée.

```
PROC OPTIONS  
group=MEMORY value;  
run;
```

Voici un extrait du résultat :

Option Value Information For SAS Option MEMSIZE  
Value: 2147483648  
Scope: SAS Session  
How option value set: Config File  
Config file name:  
C:\Program Files\SAS93\SASFoundation\9.3\nls\en\sasv9.cfg

```
PROC OPTIONS  
group=PERFORMANCE value;  
run;
```

Voici un extrait du résultat :

Option Value Information For SAS Option SPDEUTILLOC  
Value:  
Scope: Default  
How option value set: Shipped Default

Enfin, l'option TEST\_LEVEL, non documentée, peut être spécifiée au niveau de la procédure OLAP. Elle permet d'avoir des informations détaillées sur les différentes actions réalisées durant les deux phases de création d'un cube OLAP.

La syntaxe à utiliser est:

```
PROC OLAP CUBE=name TEST_LEVEL=2
<additional PROC OLAP statement options>;
Run;
```

Le format des informations affichées suit alors la logique suivante:

#### *PHASE 1*

- Connexion à SAS Metadata Server
- Initialisation de la construction du cube
- Par hiérarchie
  - Récupération des libellés de membre
  - Création des métadonnées pour les membres
  - Temps passé

#### *PHASE 2*

- Début de création du NWAY
- Ecriture du NWAY sur le disque
- Création des indexes pour le NWAY
- Début de création des autres agrégations
- Ecriture des fichiers du cube sur le disque
- Mise à jour des métadonnées du cube
- Temps passé

Toutes ces options permettent d'évaluer la situation afin de positionner les valeurs adéquates pour les différentes options de mémoire et/ou performance et de rajouter des informations complémentaires dans la log durant la création d'un cube OLAP en cas de problème.

### **Optimisation de la navigation dans un cube OLAP**

#### Prendre en compte l'environnement depuis lequel le cube OLAP est accédé

Lorsque les performances ne sont pas satisfaisantes lors de la navigation dans un cube OLAP, il est indispensable de procéder à quelques tests afin de vérifier si l'optimisation doit être effectuée au niveau du cube ou du serveur OLAP. En effet, prenons l'exemple d'un utilisateur qui accède à un cube OLAP à travers SAS® Web Report Studio depuis sa machine et où le serveur SAS (et le serveur d'applications web) est distant. Si les temps de réponse sont longs, le problème peut se situer au niveau du cube ou du serveur OLAP mais pas uniquement.

Ainsi, pour s'assurer que le problème ne provient pas du serveur d'applications web ou de la couche réseau, il est souvent intéressant de procéder à deux tests :

1. un premier test en accédant au même cube OLAP à travers un client comme SAS® Enterprise Guide® ou SAS® Add-in for Microsoft Office. Dans ces deux cas, le serveur d'applications web n'est pas utilisé et si le même problème de performance apparaît, ce composant peut être mis hors de cause.
2. un second test (soit à travers SAS Web Report Studio ou SAS Enterprise Guide / SAS Add-In for Microsoft Office selon le résultat du premier test) directement sur le serveur si possible (et non depuis un poste client) afin de s'assurer que le problème ne provient pas du réseau.

Si ces deux tests révèlent les mêmes problèmes de performances, il est probable qu'une optimisation soit nécessaire au niveau du cube, du serveur OLAP ou des deux.

#### Activer la log ARM et construire de nouvelles agrégations

Lors de la navigation dans un cube OLAP, les requêtes MDX sont exécutées de telle sorte que l'information soit recherchée dans l'agrégation la plus petite pouvant répondre au besoin.

Par défaut, seul le NWAY est construit au niveau d'un cube OLAP si bien que toutes les requêtes s'appuient sur cette agrégation pour renvoyer les résultats.

Construire de nouvelles agrégations plus petites et mieux adaptées aux requêtes exécutées par les utilisateurs permet d'améliorer considérablement les performances.

Pour ce faire, une log ARM (Application Response Measurement) est activée par défaut sur SAS OLAP Server. Elle permet de tracer et stocker toutes les requêtes exécutées par les utilisateurs sur un cube donné.

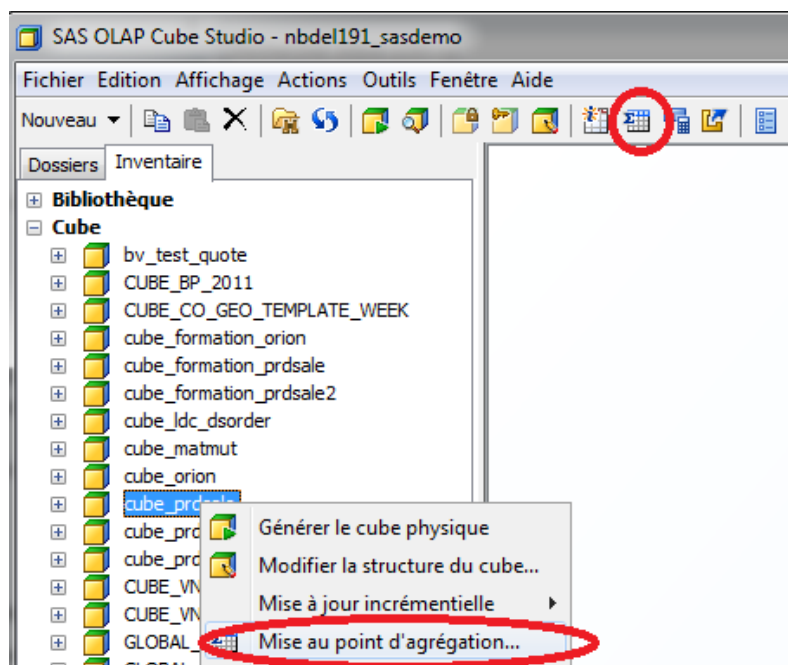
Les fichiers de log sont stockés sous la forme 'performance\_2012-04-12\_3604.arm' dans le répertoire \$SASCONFIG\Lev1\SASApp\OLAPServer\Logs où \$SASCONFIG correspond au répertoire de votre configuration SAS.

Pour plus d'informations sur la log ARM au niveau de SAS OLAP Server, vous pouvez consulter notre [documentation sur le sujet](#).

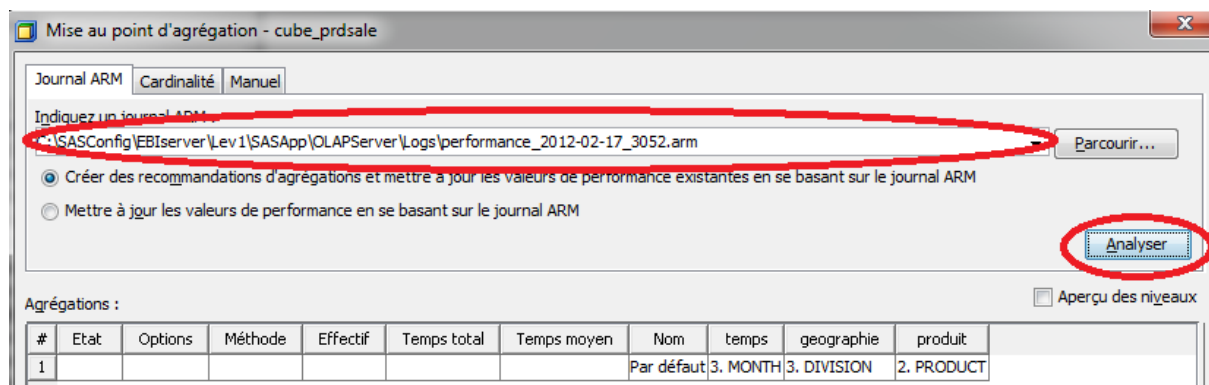
Afin d'utiliser cette log efficacement, il est tout d'abord nécessaire de se 'substituer' en tant qu'utilisateur final et de naviguer dans le cube selon les besoins exprimés par les utilisateurs.

Une fois les différentes actions effectuées, la log ARM contient alors les informations permettant d'optimiser le cube OLAP.

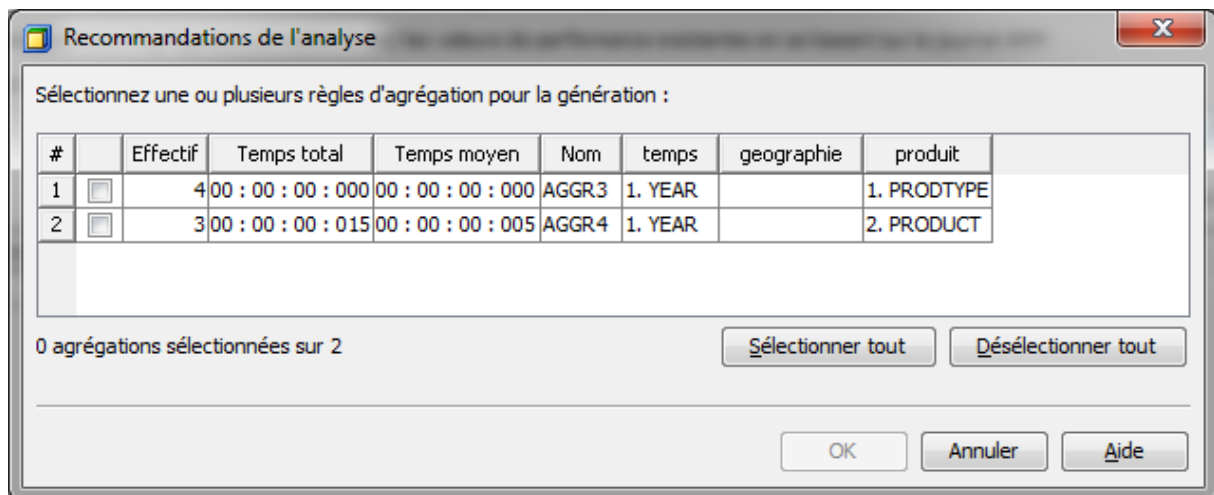
Pour analyser la log ARM générée, le moyen le plus simple est d'utiliser SAS® OLAP Cube Studio. Une fois connecté à ce produit, cliquez-droit sur un cube ou sélectionnez un cube et utilisez le bouton 'Mise au point d'agrégation', comme illustré ci-dessous :



Sélectionnez ensuite la log ARM fraîchement générée, puis cliquez sur le bouton 'Analyser' :

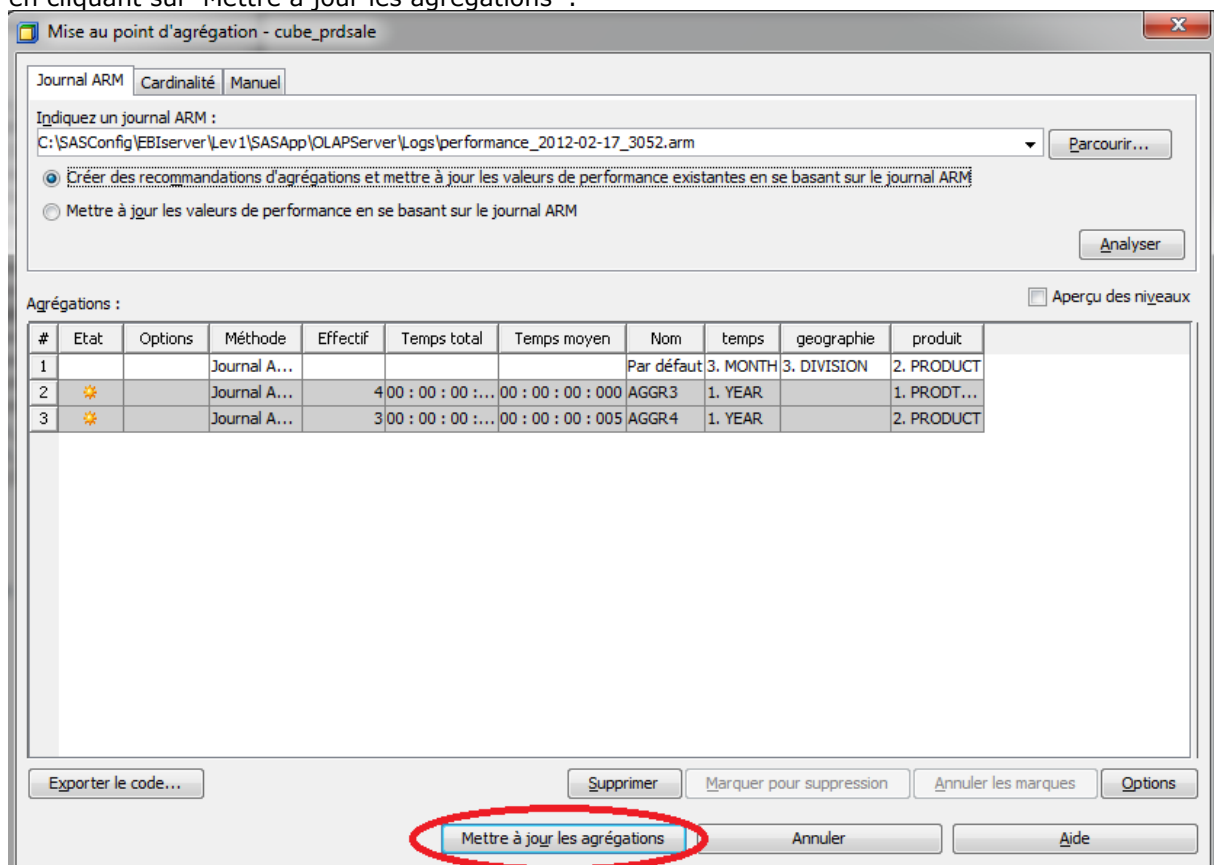


Les recommandations d'ajout d'agrégations s'affichent alors :



Cochez les différentes agrégations proposées puis cliquez sur 'OK'.

Le tableau d'agrégations se met alors à jour et il est possible de construire les agrégations proposées, en cliquant sur 'Mettre à jour les agrégations' :



Les messages dans la log permettent de vérifier que les agrégations proposées ont bien été construites :

NOTE: Processing of all cube aggregations is completed.  
 NOTE: Aggregation "AGGR3" was created with 4 records.  
 NOTE: Aggregation "AGGR4" was created with 10 records.

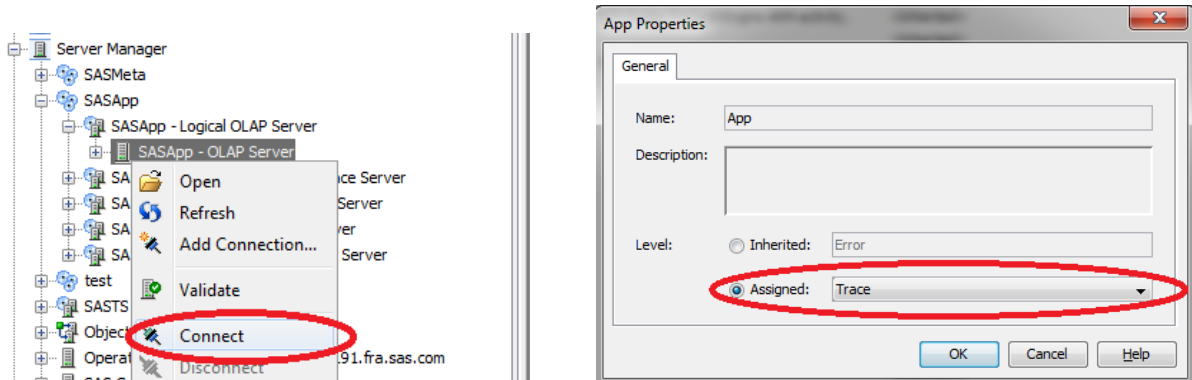
Il faut retenir que plus les agrégations créées seront proches du besoin utilisateur, meilleures seront les performances. Maintenant générer de nouvelles agrégations nécessitera bien sûr plus d'espace disque.



### Options de traces pour le serveur OLAP

Dans certains cas, il peut être compliqué d'analyser une problématique de performance lors de la navigation dans un cube OLAP, et il peut être utile d'activer des traces dans le log de ce serveur.

Dans SAS® Management Console, connectez-vous au 'SASApp - OLAP Server' avec un compte administrateur, puis positionnez le logger 'App' à 'Trace', comme illustré ci-dessous :



Vous trouverez ces informations documentées dans la Usage Note suivante :

**[Usage Note 36728: Setting additional debug options for the SAS® OLAP Server 9.2](#)**

### Options d'optimisation pour le serveur OLAP

Avant d'entrer dans le détail des options disponibles, il est important de distinguer deux types d'actions au niveau de la navigation dans un cube OLAP : la navigation 'classique' (Expand, Drilldown, etc.) et l'accès aux données de détail.

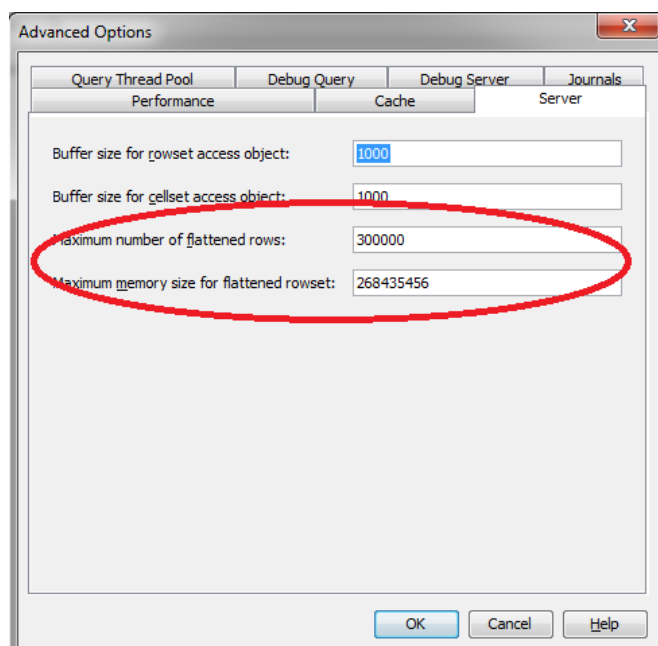
Dans le cas de l'accès aux données de détail, la requête MDX est traduite en requête SQL et exécutée sur la table/vue spécifiée pour l'accès aux détails.

Sachant cela, la première des choses à faire, si un problème de performance apparaît lors de ce type d'action, est d'optimiser la table/vue en créant des indexes.

Il est également possible de limiter le nombre de lignes retournées et la mémoire allouée au serveur OLAP pour l'accès aux données de détail via les options ci-dessous :

- Maximum number of flattened rows
- Maximum memory size for flattened rowset

Ces options sont disponibles à travers SAS Management Console, dans les propriétés avancées du SASApp - OLAP Server, dans l'onglet 'Server' :



Attention, le nombre de maximum de lignes retournées pour l'accès aux données de détail est positionné à 300 000 par défaut. Il est possible d'augmenter cette valeur mais dans ce cas, la valeur de '*Maximum memory size for flattened rowset*' devra alors également être adaptée.

Si la structure de la table de détails est très importante (nombre important de colonnes), la mémoire nécessaire pour restituer un faible nombre de lignes peut être élevé. Dans ce cas, nous conseillons de diminuer la valeur de l'option '*Maximum number of flattened rows*' afin de restituer moins de lignes et ainsi diminuer la mémoire requise pour la restitution.

Si des erreurs de type « Out of Memory » ou « Insufficient Memory » apparaissent, les mêmes options pourront être manipulées.

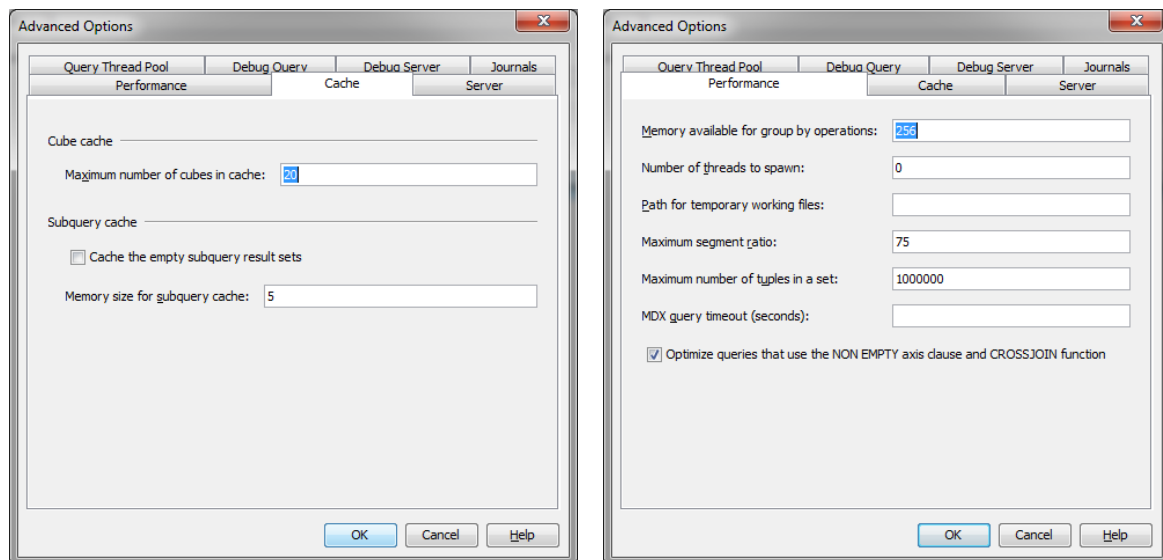
Après avoir traité le cas de l'accès aux données de détails, voyons maintenant les actions de navigation classiques dans un cube OLAP.

Si l'ajout de nouvelles agrégations ne donne toujours pas satisfaction en termes de temps de réponse, certaines options du serveur OLAP peuvent adresser ce besoin.

La première option à positionner est l'option MEMSIZE qui permet d'allouer la mémoire disponible pour le serveur OLAP.

Cette option peut être positionnée dans le fichier sasv9.cfg associé au serveur OLAP.

Comme pour l'accès aux données de détails, certaines options peuvent également être positionnées à travers SAS Management Console au niveau des options avancées du SASApp – OLAP Server, cette fois dans les onglets 'Cache' et 'Performance'.



Augmenter le nombre de cubes dans le cache peut considérablement améliorer les temps de réponse mais nécessite alors d'allouer plus de mémoire pour le serveur OLAP.

Si votre cube a un nombre relativement élevé de calculs par requête, que le nombre de requêtes concurrentes sur le cube est relativement bas et que le cube n'a pas trop de cellules vides, l'option 'Cache the empty subquery result sets' peut alors être activée.

Enfin l'option 'Optimize queries that use the NON EMPTY axis clause and the CROSSJOIN function' est activée par défaut au niveau du serveur OLAP afin d'optimiser les requêtes MDX basées sur les fonctions NON EMPTY et CROSSJOIN.

En cas de problème de performance, il faut toutefois vérifier que cette option est toujours activée.

## **Conclusion**

Ce document vous a présenté les principaux cas d'optimisation des cubes OLAP lors de la création et la navigation. Il faudra donc retenir qu'une optimisation peut/doit être effectuée au niveau du cube mais également au niveau du serveur OLAP.

Outre toutes les options présentées dans cet article, d'autres solutions au niveau de l'architecture existent afin d'améliorer les temps de réponse.

Il sera par exemple possible de mettre en place un cluster au niveau du serveur OLAP afin de répartir la charge entre les différentes instances OLAP.

Il sera également possible de disposer de plusieurs instances OLAP distinctes afin de dédier l'utilisation de tel ou tel cube à une instance OLAP bien précise.

### Liens utiles

- Partie [Tuning SAS OLAP Servers with Advanced Server Options](#) dans la documentation SAS 9.3 Intelligence Platform: Application Server Administration Guide
- Options de la procédure OLAP dans la documentation SAS 9.3 OLAP Server: User's Guide: [The OLAP Procedure](#)
- [Optimizing the cube build process in SAS® 9.2](#)
- Partie "Implement recommended best practices for OLAP querying" dans le document [Usage Note 39065: Documented best practices and other resources for improving the performance of SAS® Web Report Studio](#)

Sylvain BERTIN  
Consultant Support Clients SAS France