



ADMINISTRATION BASE DE DONNEES

SYBASE ASE 12.5

Complément de cours

Introduction : Adaptive Server Enterprise

Adaptive Server Enterprise (ASE) est un système ultraperformant de gestion de bases de données relationnelle destiné aux environnements critiques à forte volumétrie.

Les systèmes de traitement de volumes extrêmement importants de transactions doivent prendre en charge des dizaines de milliers d'utilisateurs simultanés, tout en garantissant la rapidité et la continuité des transactions sur des plates-formes rentables et standard. ASE répond à toutes ces exigences et plus encore. Reconnu pour sa fiabilité hors du commun, ses performances élevées et son faible coût total de possession, le système critique de gestion de bases de données d'entreprise ASE équipe 24 des 25 premières banques et 46 des 50 premières banques et sociétés de courtage mondiales.

ASE intègre principalement une technologie de cryptage et de partitionnement **brevetée**, ainsi qu'une technologie de requête pour garantir des transactions « **plus intelligentes** », ainsi qu'une disponibilité continue des environnements en clusters. Grâce à l'intégration de la technologie des **bases de données en mémoire** (ASE 15.5), il est possible de virtualiser des données et d'offrir la capacité d'évolution requise pour répondre aux besoins des entreprises devant gérer d'importants volumes de données et un grand nombre d'utilisateurs simultanés, qu'elle soit déployée dans des environnements de cloud public ou de datacenters privés.

Performances exceptionnelles	<ul style="list-style-type: none">• Permet aux environnements informatiques exigeants de traiter des millions de transactions par minute à l'aide de bases de données d'un téraoctet, tout en maintenant des taux de croissance élevés du volume des données et des transactions.
Risques réduits	<ul style="list-style-type: none">• Permet aux entreprises d'éviter les perturbations désastreuses et coûteuses, telles qu'une défaillance système, des erreurs imprévues ou des vols de données.• Protège contre les menaces courantes qui pèsent sur les systèmes informatiques, notamment pour la sécurité des données, la stabilité du système et la reprise après sinistre.• Permet aux équipes informatiques d'offrir une assistance technique pertinente et d'atténuer les risques.
Efficacité accrue	<ul style="list-style-type: none">• Permet aux équipes informatiques de fournir des niveaux de service élevés, même avec des budgets restreints.• Utilise efficacement les équipements et élimine ceux en sous-utilisation, offrant ainsi des performances élevées avec des dépenses totales moindres.• Simplifie les tâches de base de données requises.

(Source : sybase.fr)

I. Introduire ASE 12.5 :

Un administrateur système est responsable de la création, la gestion et la maintenance des serveurs de base de données. Ces derniers se présentent comme un environnement complet, et, en perpétuelle interaction.

1. Installation ASE :

A- Bases de données obligatoires :

Avant d'installer Sybase ASE 12.5, il est important de tenir compte de l'espace nécessaire pour les bases de données obligatoires et éventuellement celui des bases de données optionnelles et celles utilisateurs.

Les bases de données obligatoires sont :

Master : Contient les tables systèmes qui enregistrent les données relatives à la gestion de tout le système.
Exemple : *sysdatabases, syslogins, sysdevices,...* (**Taille min 6MB**)

Model : Une base template qui est utilisée pour créer les bases de données utilisateurs. Tous les objets : tables, procédures, triggers, rôles, règles,... créés dans modèle, seront reproduits dans toute base de données utilisateur. (**Taille min 2MB/Master device**).

Sybsystemprocs : Contient les tables relatives aux procédures stockées système (**taille min 100MB**).

Tempdb : Contient les tables temporaires (**Taille min 2MB/Master device**). La taille recommandée est le max de : l'espace totale demandé lors de l'exécution du plan le plus large ; 10% de la taille des données ou 100 M ; 1 à 2 M par connexion utilisateur.

B- Bases de données additionnelles (optionnelles) :

Sybsyntax : Contient une aide de syntaxe des commandes T-SQL qu'on obtient avec la commande

```
sp_syntax "keyword"
```

Dbccdb : contient les entrées sorties de la commande **dbcc checkstorage**

Sybsecurity : Contient des informations permettant de faire un audit sur les utilisateurs

Bases de données exemple : pubs2, pubs3, ...

C- Installation d'ASE :

La première tâche dont un SA est responsable est l'installation du Serveur de base de données. Cependant, le choix de la version, du système d'exploitation et toute autre ressource qui constitue l'environnement présente un vrai dilemme.

Un des points forts de Sybase et sa modeste consommation des ressources : mémoire, temps CPU, Stockage,...

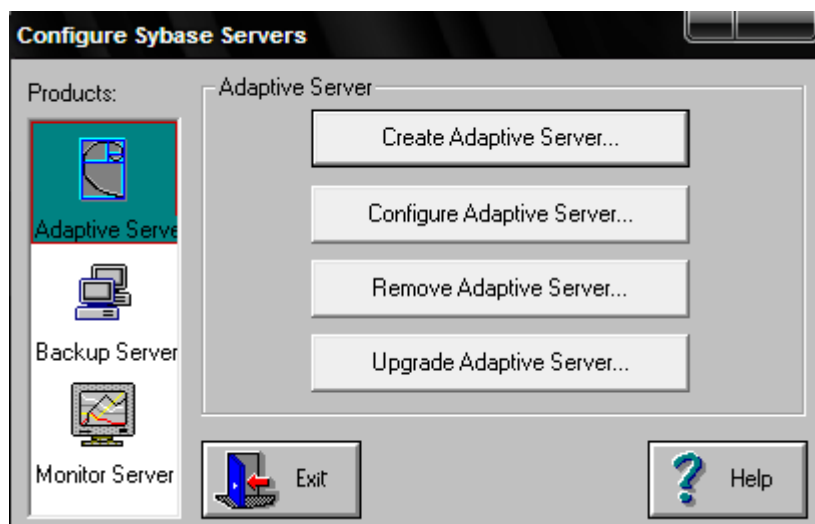
(Comparatif SGBDR : <http://fadace.developpez.com/sbgdcmp/>)

Ceci dit, voici une liste des systèmes d'exploitation sur lesquels on installerait ASE 12.5 :

Solaris 2.8, 32 & 64 Bits
HP/UX 11.1, 32 & 64 Bits
IBM AIX 4.3.3, 32 & 64 Bits
Intel NT 4 SP 6
Digital Unix TruUnix 5.0a 64 bits
SGI Irix 6.5, 32 & 64 Bits
Intel Linux Kernel 2.2.14-50 glibc 2.1.3-15
Windows NT x86 / x64
MAC OS

L'espace minimal requis par l'installation varie entre 400 et 800 MO dépendant de la plate-forme utilisée et des options d'installation choisies.

Une fois toutes les composantes d'ASE installées, on procédera à la configuration du serveur de base de données. Sous Windows, ceci se fait par le biais de l'utilitaire : **[Server Config]**



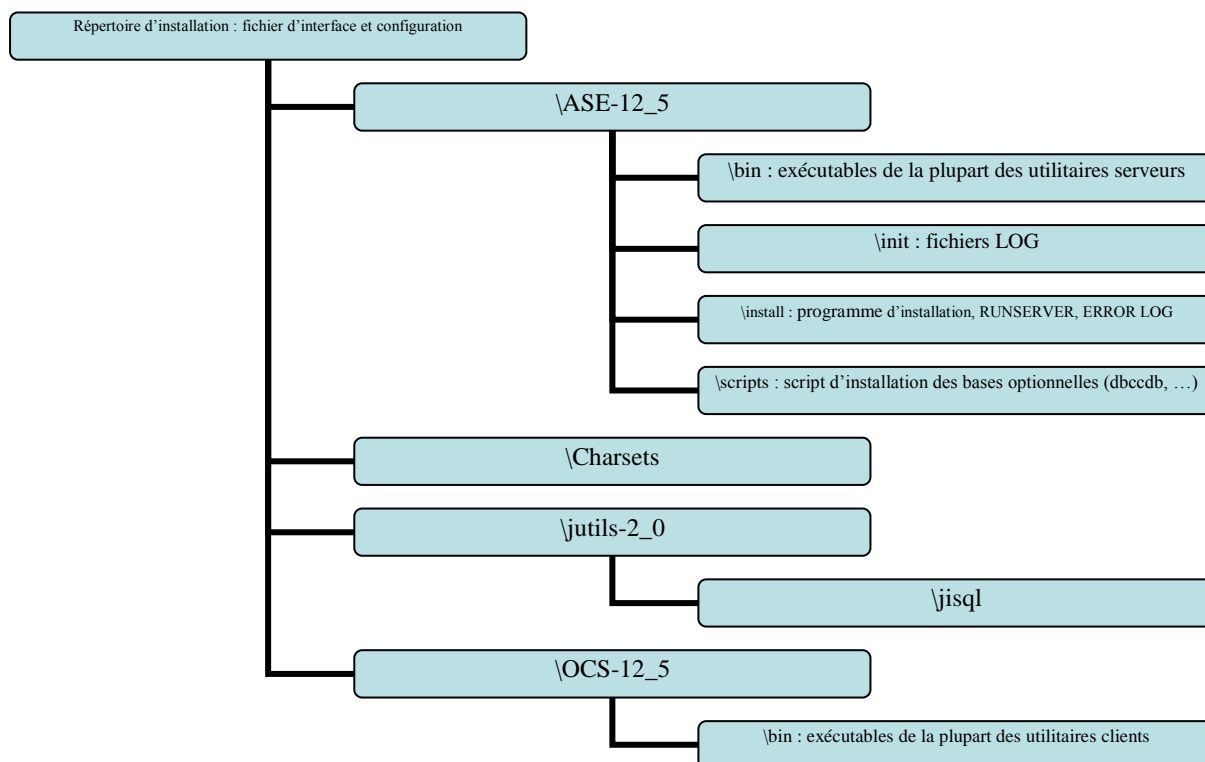
Sous Unix, on utilisera la commande :

srvbluidres [-v] [-rfilename]

-v : Imprime la version et les informations de copyrights

-rfilename : Précise le fichier de ressource à utiliser pour créer le serveur. Un fichier de ressource exemple est présent dans : \$SYBASE/ASE-12_5/init/

2. Structure des répertoires :



3. Démarrage et arrêt du serveur :

On sera souvent amené à démarrer un des serveurs de base de données, de le stopper et de vérifier ceux en cours d'exécution. Ceci émane, bien sûr, des différentes tâches d'administration nécessitant ce type de manœuvre.

La création d'un serveur, ou la configuration d'un serveur, génère un fichier de démarrage qui porte le nom du serveur et qui se localise dans le répertoire : Install. En s'y positionnant, on lancera la commande prompt suivante :

Startserver -f[m] RUN_servername[.bat]

-m : lancera le serveur en mode mono-utilisateur. Dans ce cas, le paramètre de configuration « **allow updates to system tables** » est activé permettant la mise à jour des tables systèmes par des requêtes utilisateurs (Faites très attention car une mauvaise manipulation peut endommager tout le système).

Pour vérifier les serveurs démarrés :

Showserver

Pour arrêter, on lancera la commande SQL (depuis SQLAdantage ou isql,...)

Shutdown [server_name] [with {wait | nowait}]

Notez qu'un service Windows est créé pour chaque serveur.

4. Connexion au serveur :

Les connexions au serveur sont gérées par un fichier d'interface qui peut être défini comme étant un « livre d'adresse » qui spécifie la localisation des serveurs et les ports d'écoute et d'émission.

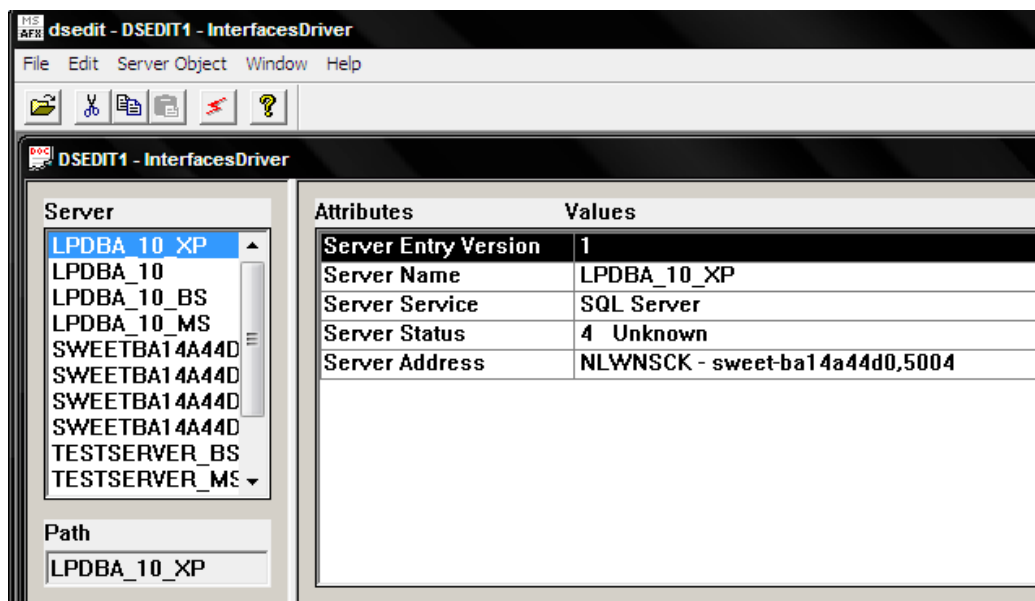
Sous Windows, on l'appellera « sql.ini ». Chaque serveur est représenté par une entrée dans ce fichier sous la forme :

[SERVER_NAME]

Master = NETWORK_PROTOCOLE, SERVERHOST_NAME | Adresse IP, PORT_NUMBER

Query = NETWORK_PROTOCOLE, SERVERHOST_NAME | Adresse IP, PORT_NUMBER

On pourra toute fois utiliser l'utilitaire : [dsedit]



Exemple :

[LPABD_11]

master=NLWNSCK, 172.16.1.1, 5000

query=NLWNSCK, 172.16.1.1, 5000

Sous Unix, on l'appellera « interfaces ». Les entrées serveurs prendront la forme :

SERVER_NAME RETRY DELAY

QUERY PROTOCOLE ETHER HOSTNAME | ADRESSE_IP PORT_NUMBER

MASTER PROTOCOLE ETHER HOSTNAME | ADRESSE_IP PORT_NUMBER

Exemple :

LPABD_11 3 10

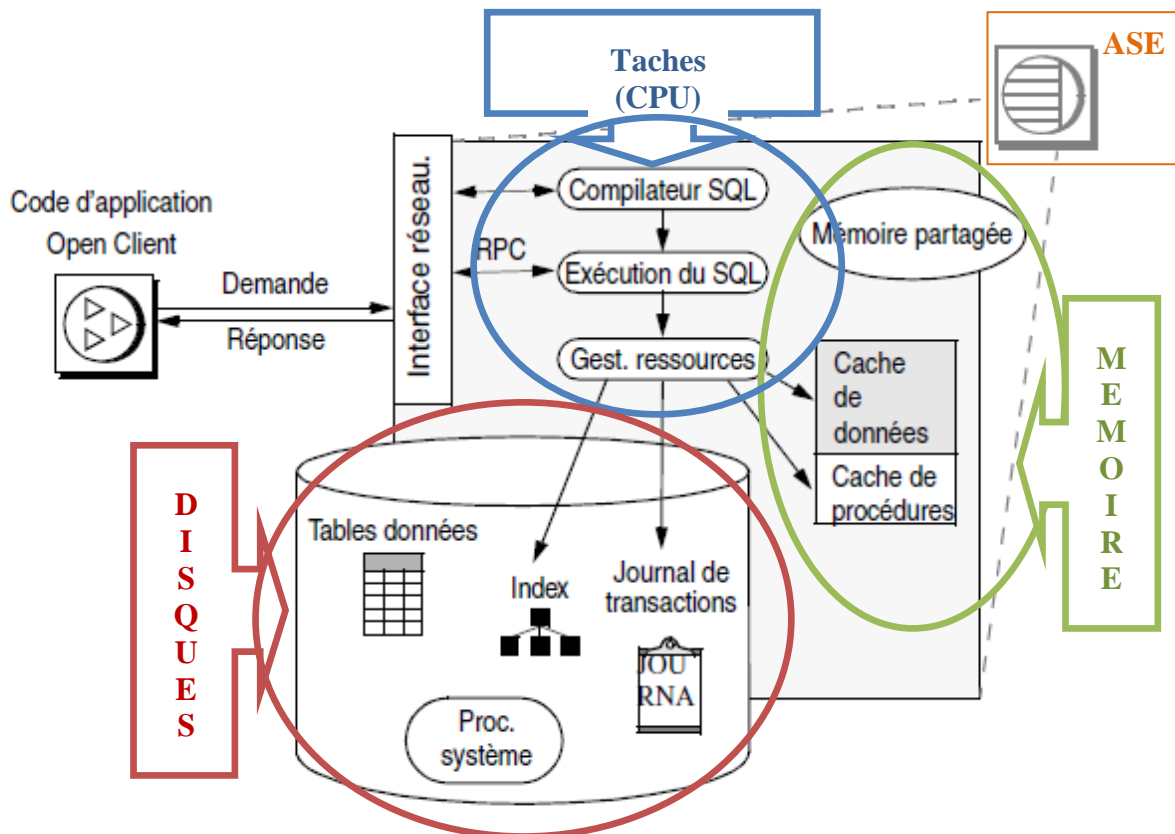
Query tcp ether SERVER_CLASSE 5000

Master tcp ether SERVER_CLASSE 5000

II. Configurer ASE 12.5 :

1. Fichier de configuration :

Comme nous l'avons introduit, le serveur de base de données ASE 12.5 se présente comme un environnement à part entière. Le schéma suivant illustre brièvement les composantes de cet espace :



Somme toute, ASE a besoin d'identifier, de gérer, d'optimiser chaque composante. Pour cela, on recourt au fichier de configuration qui contient les paramètres généraux de configuration de l'ASE et de son environnement. Ces informations sont stockées dans un fichier nommé : SERVERNAME.cfg.

Notez qu'il existe deux types de paramètres : statiques, qui nécessitent le redémarrage du serveur pour qu'ils prennent effet, et dynamiques.

Au démarrage, ASE crée une copie du fichier de configuration qu'il nommera SERVERNAME.bak. Cette copie sera écrasée et remplacée par une autre au prochain démarrage du système. Cependant, chaque changement d'un paramètre est tracé par la création d'une copie de sécurité qu'on nommera SERVERNAME.nnn, où nnn sera entre 001 et 999. Cette copie de sécurité servira lors d'un éventuel problème dû à une mauvaise configuration. Faites tout de même attention aux changements directs en éditant en mode texte le fichier de configuration et pensez vous-même à créer une copie !

Pour lister tous les paramètres : **sp_configure**

Pour lister un groupe de paramètres : **sp_configure "groupe_name"**

Pour lister un seul paramètre : (sp_helpconfig affiche les mêmes infos que sp_configure en y ajoutant la valeur min et max que peut prendre le paramètre)

sp_configure "parameter_name" ou sp_helpconfig "parameter_name"

Pour modifier (notez qu'on peut l'éditer manuellement) : **sp_configure "parameter_name", parameter_value**

Pour revenir à la valeur par défaut : **sp_configure "parameter_name",0,"default"**

Pour écrire les valeurs configurées et actives pendant l'exécution d'ASE dans un fichier :

Sp_configure "configuration file", 0, write, "c:\...\new_file.cfg"

Pour écrire les valeurs configurées et pas nécessairement actives pendant l'exécution d'ASE dans un fichier :

Sp_configure "configuration file", 0, restore, "c:\...\new_file.cfg"

Pour vérifier un fichier :

Sp_configure "configuration file", 0, verify, "c:\...\new_file.cfg"

Pour vérifier les paramètres d'un fichier modifié manuellement et charger ceux qui passent : (attention, les valeurs des paramètres statiques doivent être les mêmes que celles en exécution)

Sp_configure "configuration file", 0, read, "c:\...\new_file.cfg"

Pour savoir quel fichier est utilisé :

Select value2 from master..syscurconfigs where config=14

Tables systèmes qui contiennent les informations sur le fichier de configuration sont :

sysconfigures : contient les valeurs des paramètres depuis le démarrage ou depuis la dernière modification

syscurconfigs : contient les valeurs utilisées des paramètres.

2. Configuration de la mémoire dans ASE :

Un des éléments clés de la performance est la mémoire. Quoique Sybase gère ce côté avec une grande sobriété, il est toute fois critique de choisir la bonne philosophie de gestion et de paramétrage surtout dans de grands systèmes.

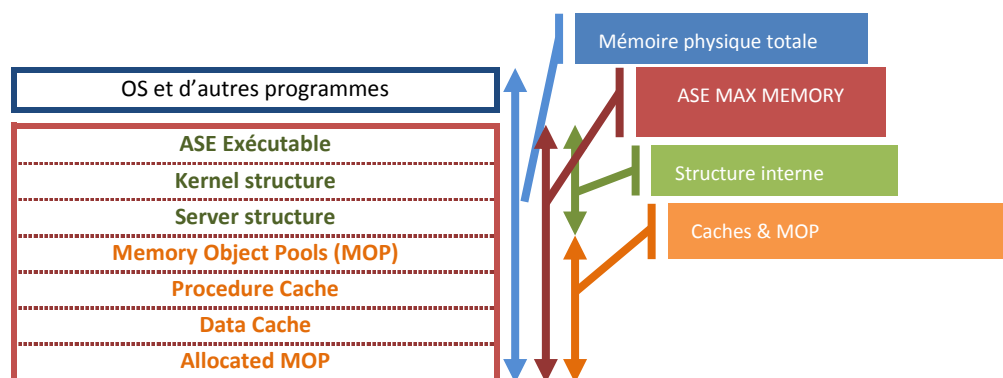
Avoir beaucoup de mémoire réduit considérablement les E/S disques ce qui améliore les performances et diminue le temps de réponse : premier souci de l'utilisateur final !

Dans l'ensemble, la mémoire est consommée en grande partie par les DATA CACHES, parties de la mémoire où seront chargées les pages de données, PROCEDURE CACHE et les différents paramètres qui en auront besoin.

Les points clés à retenir sont :

- La taille maximale de la mémoire à configurer pour ASE.
- La philosophie d'allocation de la mémoire : Dynamique ou statique.
- La valeur optimale pour chaque paramètre consommant de la mémoire.
- Les petites proportions allouées aux différentes tâches ne seront jamais négligés du fait de la « répétabilité » et la concurrence des différents utilisateurs !!!!

L'architecture de la mémoire se présente comme suit :



MAX MEMORY :

La mémoire physique totale de l'ASE exprimée en page. Exemple en 2K pour 5000KO :

```
sp_configure "max memory", 2500
```

ASE Exécutable & Kernel:

Non configurable

Server structure :

Configurable en paramétrant les métas-data:

number of user connections

number of open databases

number of open indexes

number of open objects

number of locks (recommendation : (concurrent user connection + worker process)x20)

number of devices.

Number of aux scan descriptors : un descripteur est utilisé pour chaque table référencée, directement ou indirectement, dans une requête utilisateur. Si, par exemple une table est référencée deux fois dans la même requête, deux descripteurs seront alors utilisés. On attribue 48 descripteurs pour chaque connexion utilisateur : 16 pour les tables utilisateurs, 12 pour les tables temporaires, 16 pour les tables systèmes et 4 pour les conditions de ROLL BACK. Si une requête à besoin de plus de descripteurs, elle utilisera alors les descripteurs auxiliaires, si elle en trouve pas elle s'arrêtera !

Pour estimer la proportion de la mémoire souhaitée :

```
sp_helpconfig "parameter_name", "size"
```

```
sp_countmetadata "parameter_name"
```

```
sp_monitorconfig "parameter_name"
```

Exemple:

```
sp_helpconfig "number of user connections", "100"
```

Configuration parameter, 'number of user connections', will consume 8601K of memory if configured at 100.

...

Ou:

```
sp_helpconfig "number of user connections", "50M"
```

Configuration parameter, 'number of user connections', can be configured to 579 to fit in 50M of memory.

MOP :

Parties de la mémoire allouées selon la philosophie choisie par l'administrateur. Cette stratégie dépendra de deux paramètres clés : *Allocate max shared memory* et *dynamic allocation on demand*. Quand le premier est configuré à 1, toute la mémoire est allouée au démarrage d'ASE et aucune partie libre ne sera déclarée. Dans le cas contraire, juste la mémoire nécessaire pour satisfaire les différents paramètres et caches sera allouée au démarrage. Si vous changez les paramètres en exécution d'ASE, l'affectation de la mémoire suivra alors deux autres logiques : si le 2^{ème} paramètre est configuré à 1, on n'allouera la mémoire qu'en cas de besoin, sinon, on le satisfera complètement dès le changement.

Total logical memory nous renseignera sur la mémoire totale nécessaire pour ASE, *Total physical memory* reportera par contre la mémoire réellement allouée par ASE.

Les paramètres les plus importants à configurer sont :

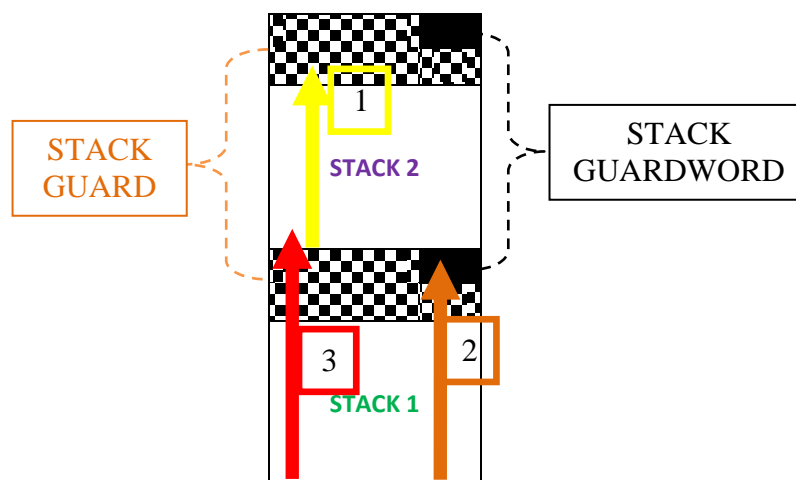
Number of user connections Nombre de connexions concurrentes.

Heap memory per user : Heap memory est une pile de mémoire interne que les tâches, interrogeant de grande colonne, utilisent dynamiquement pour être réalisées. Sa taille totale est obtenue par la variable globale **@@heapmemsize** ou calculée par la formule :

$$(1024 \times 1024) + (\# \text{ user concurrent} + \text{Worker process}) \times \text{HMPU} \times 1,05 \sim 2K$$

Stack size : une zone de mémoire où les processus utilisateurs gardent la trace du déroulement, de leur contexte et y stockent leur données internes.

Stack guard size : Du fait que les stacks constituent une chaîne contigus, le risque, qu'un processus « viole » une stack d'un autre processus, est présent. Pour remédier à cette éventuelle situation, ASE recourt à une zone de protection : STACK GUARD AREA qu'une fois un processus la franchit (1) ASE l'arrête avant qu'il pénètrerait la stack suivante. Cette recherche est périodique, du fait, un processus peut ne pas être « capturé » dans la zone de protection. On ajoute alors une GUARDWORD qui marque effectivement la barre à ne pas franchir. S'il est atteint (2), ASE s'arrête complètement et enregistre une erreur dans le LOG général. Le cas échéant (3) est le dépassement réel de la STACK GUARD AREA vers la stack suivante provoquant là aussi l'arrêt définitif d'ASE.



Default network packet size : la taille par défaut (utilisée dans la majorité des cas) des paquets réseaux. Ce paramètre est déterminant pour la taille globale de la mémoire réservée pour le réseau :

$$\text{Mémoire réseau} = (\# \text{ user concurrent} + \text{Worker process}) \times 3 \text{ Buffers} \times \text{DNPZ}$$

Max network packet size : la taille maximale que peut prendre un paquet réseau. Elle doit être supérieure ou égale à la DNPS. Si un processus tente d'utiliser un paquet de taille supérieure à la DNPS et que la MNPS ne le permet pas, une erreur sera générée. Ceci dit, pour que cela soit faisable, déjà, ASE a besoin d'une mémoire additionnelle qu'on paramètre via **additional network memory**. La taille de la mémoire réseau additionnelle est estimée par la formule :

$$(\text{Somme des paquets simultanés} > \text{DNPZ}) \times 3 \text{ Buffers} \times 1,2 \sim 2048$$

isql -A2048 forcera l'utilitaire isql à travailler avec un paquet réseau de 2K.

Max roles enabled per user : nombre de rôles activés par un utilisateur. Notez que le nombre maximal de rôles que peut activer un utilisateur est 127 et que le nombre maximal de rôle à activer dans ASE est de 992 dont ceux (32) systèmes.

User log cache size : la taille du « petit journal » créé pour chaque utilisateur où ses transactions seront journalisées, servant à diminuer la contention sur le journal principale. Si les utilisateurs utilisent des grandes transactions, et si la taille du cache utilisateur n'est pas suffisante, les traitements seront abandonnés.

Procedure Cache :

Configurable = Max user connection x size of largest plan x 1.25 exprimé en taille de page. Pour obtenir la taille du plus grand plan on exécute la requête suivant :

```
select max(count(*)/8+1) from sysprocedures group by id
```

Pour le configurer - exemple: **sp_configure "procedure cache size", 7000**

Data cache :

Charge les pages de données, d'indexe et de log.

(Les caches de données et de procédures feront l'objet d'un chapitre complet)

III. Ressources physiques :

1. Devices :

Un **DEVICE** est une ressource physique qui peut être un ou plusieurs disques ou une portion ou plusieurs de disque. La taille max d'un device est 32GB. La taille min dépend de la taille de page : 2K : 1MB ; 4K : 1M ; 8K : 2MB ; 16K : 4M. Les informations sur les devices sont enregistrées dans la table système sysdevices.

Commandes relatives aux devices :

disk init : initialise les devices et les crée (la clause **dsync** détermine si oui ou non l'écriture est faite directement sur le disque sous UNIX seulement)

sp_dropdevice : supprime un device. Pour faire, on désalloue toutes les bases de données qui y sont liées. Cette commande ne supprime pas physiquement le fichier.

sp_helpdevice : affiche les informations sur tous les devices ou sur un de précis.

sp_diskdefault : change le device par défaut – exemple : **sp_diskdefault dev_data1, defaulton**

sp_deviceattr : change un paramètre de device – exemple : **sp_deviceattr, dsync, false**

disk mirror : active le mirroring.

disk remirror : reactive le mirroring.

disk unmirror : désactive le mirroring.

Concernant le mirroring, si le device principale tombe en panne, ASE le désactive et continue avec le deuxième et change le statut du premier pour marquer son arrêt. On pourra créer une alerte automatique à l'aide de WAITFOR

Exemple:

```
Begin
    Waitfor mirrorexit
    Exec Proc_notifie_DBA_UNMIRROR
End
```

2. Databases :

Selon la définition de Wikipédia, une base de données est un lot d'informations stockées dans un dispositif informatique. Les technologies existantes permettent d'organiser et de structurer la base de données de manière à pouvoir facilement manipuler le contenu et stocker efficacement de très grandes quantités d'informations.

ASE 12.5 peut gérer jusqu'à 32767 bases de données à la fois. Ainsi, pour créer une il faut se positionner dans la base de données master (use master) et que vous avez déjà le privilège d'en créer.

Estimer la taille éventuelle que va avoir votre base de données dans un an, 2 ou même 10 est une opération très délicate. La procédure **sp_estspace** permet de quantifier la taille d'une table par rapport à un nombre éventuel d'enregistrement, et optionnellement la valeur du **fill_factor**, les colonnes qui prennent la taille maximale de leur type déclaré, la taille des champs LOB et le nombre des E/S disques.

```
/* La taille que prendra titles si elle aura 10000 lignes*/
sp_estspace titles, 10000
```

```
-----
Total_Mbytes
-----
1.94
```

```
/* La taille que prendra titles si elle aura 10000 lignes avec fill_factor = 75%*/  
sp_estspace titles, 10000,75
```

```
-----  
Total_Mbytes
```

```
-----  
2.59
```

create database : avec l'option « with override » si vous précisez « log on » sur le même device des données.

drop database

sp_helpdb

sp_changedbowner : pour changer le propriétaire de la base de données.

sp_dboption : configure les options de la base de données. Avec checkpoint pour que les changements prennent effet.

Alter database : pour changer un élément de la base de données.

sp_logdevice : pour convertir un nouveau device lié à la base de données en un device log. La recommandation pour la taille du log est : 10% à 25% de la taille de la base de données.

sp_configure "default database size", 4 change la taille par défaut (en MB) des bases de données. La taille minimale est 1M x Logical page size.

db_name(), db_id() : renvoi respectivement le nom depuis l'id de la base, l'id depuis le nom de la base.

grant create database to : pour donner le droit de création des bases à un login. Cet utilisateur ne doit pas être un de la base de données en question.

Options des bases de données (sp_dboption) :

allow nulls by default : si à 1, permet d'attribuer la propriété null à une colonne dont la propriété n'a pas été mentionnée lors de la création d'une table.

auto identity : si à 1, ajoute une colonne d'identité de 10 chiffres à une table qui n'a pas de clé primaire à la création. Notez que cette colonne ne figure pas dans un select *. Pour la faire apparaître : select SYB_IDENTITY_COL. Pour changer sa taille (10 par défaut) changer le paramètre de configuration ASE : **size of auto identity**.

dbo use only : la base ne peut être utilisée que par son propriétaire.

ddl in tran : si à 1, les commandes create, drop et alter peuvent être lancées depuis une transaction définie par utilisateur (user_defined transactions). Les commandes suivantes ne peuvent jamais être lancées depuis un user_defined transactions : **alter database, alter table ... (un)partition, create database, disk init, dump database, dump transaction, drop database, load transaction, load database, select into, truncate table, update statistics**.

identity in nonunique index : si à 1, quand l'option auto identity est à true, et que la table a une colonne d'auto-identité, celle-ci va être ajoutée automatiquement à un index qui n'est pas unique.

read only : si à 1, la base est en lecture seule.

single user : seulement un seul utilisateur peut accéder à la base (le premier).

unique auto_identity index : si à 1, ajoute une colonne unique, si elle n'existe pas, avec un index unique non clusterisé à la création d'une table.

select into/bulkcopy/plsort : si à 1, autorise les opérations non journalisées complètement.

no free space acctg : si à 1, seulement les thresholds sur le logsegment sont actives, à 0, toutes les thresholds sont actives.

abort tran on log full : si à 1, le serveur abandonne toutes les transactions ouvertes si le log est rempli. Si à 0, il les suspend.

no chkpt on recovery : si à 1, l'enregistrement qui présente un checkpoint après le recovery d'une base n'est pas ajouté au journal.

trunc log on chkpt : si à 1, le journal des transactions est tronqué (transactions validées sont supprimés) lors d'un checkpoint si seulement plus de 50 lignes sont écrites dans le journal.

Tables systèmes en jeu : sysdatabases, sysusages (enregistre les fragments d'une base)