

Université **IBM i**

7 novembre 2023

IBM Innovation Studio Paris

S07 – Contrôler l'exécution des requêtes SQL avec Query Supervisor et QAQQINI

13:30 / 14:30

Nathanaël Bonnet

Gaia/Volubis

nathanael.bonnet@gaia.fr

 **infrasdufutur**

#ibmi

#uui2023

#infrastructuredufuturIBM23



Infrastructures du futur



7 et 8 novembre 2023

Présentation

- Nathanaël BONNET
 - IBM i depuis 1999
 - Expert IBM i
 - Développement & intégration



Agenda



- 1. Introduction
 - Rappels
 - Périmètre

- 2. Query Supervisor
 - Usage

- 3. QAQQINI
 - Tuning
 - DB2 SMP



Infrastructures du
futur

7 et 8 novembre 2023

Université **IBM i**

7 novembre 2023

1. Introduction



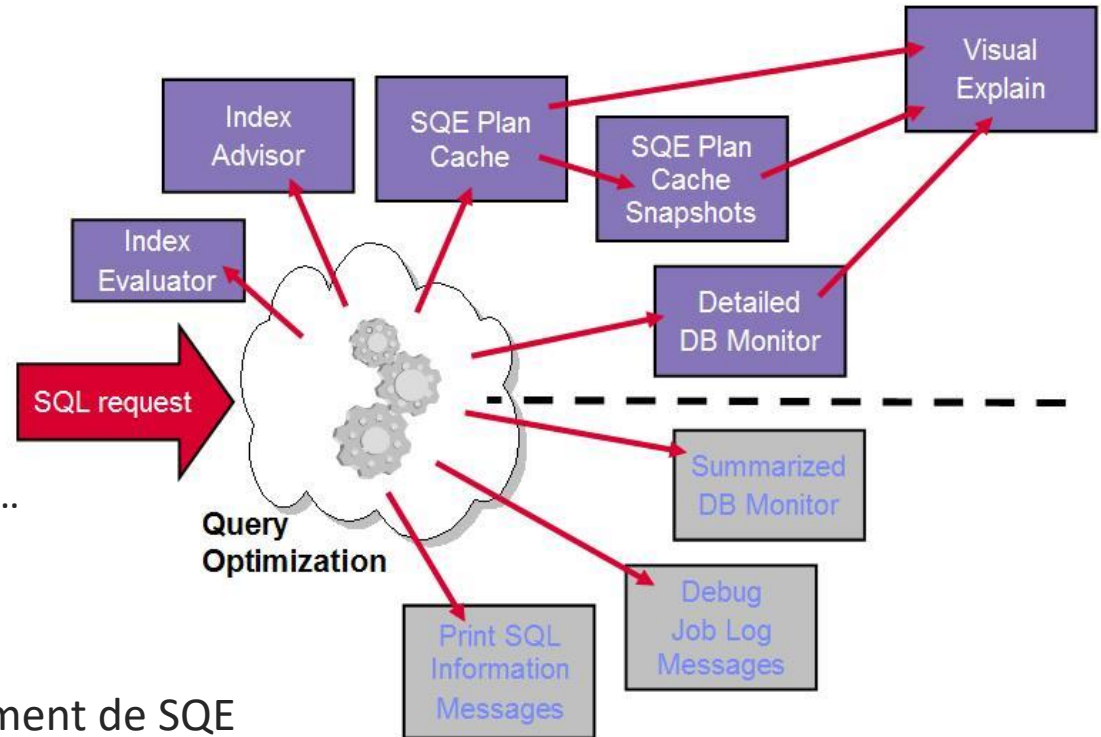
Let's
Create

Nous ne verrons pas

- Performances SQL
 - ➔ Voir S13 - Bonnes pratiques pour écrire un code SQL efficient – 14h45 à 15h45
- La mise au point extrême des requêtes, au-delà de l'outillage
 - ➔ Voir S21 - L'hyper optimisation sous DB2 for i – 16h à 17h – Christian Grière
- Query Governor, moniteur de base de donnée
 - ➔ Cf historique université

Rappels

- Exécution d'une requête SQL
- Avec IBM i 7.4 TR4 / 7.3 TR10
 - Query Supervisor
 - Permet de fixer des seuils
 - Interruption des requêtes
 - De façon globale
 - Ou par travail, utilisateur, ...
- QAQQINI
 - Permet de fixer le comportement de SQE





Infrastructures du
futur

7 et 8 novembre 2023

Université **IBM i**

7 novembre 2023

2. Query Supervisor



Let's
Create

Query Supervisor

- Disponible avec IBM i 7.3 TR10 et IBM i 7.4 TR4
 - Permet de surveiller l'exécution des requêtes SQL
 - Référence : <https://www.ibm.com/docs/en/i/7.5?topic=tools-query-supervisor>

- Permet de définir des seuils
 - CPU
 - Temps
 - Stockage temporaire
 - Opérations d'I/O

- Lorsque le seuil est atteint
 - Déclenchement d'une action (un programme)
 - Via point d'exit QIBM_QQQ_QRY_SUPER

Query Supervisor

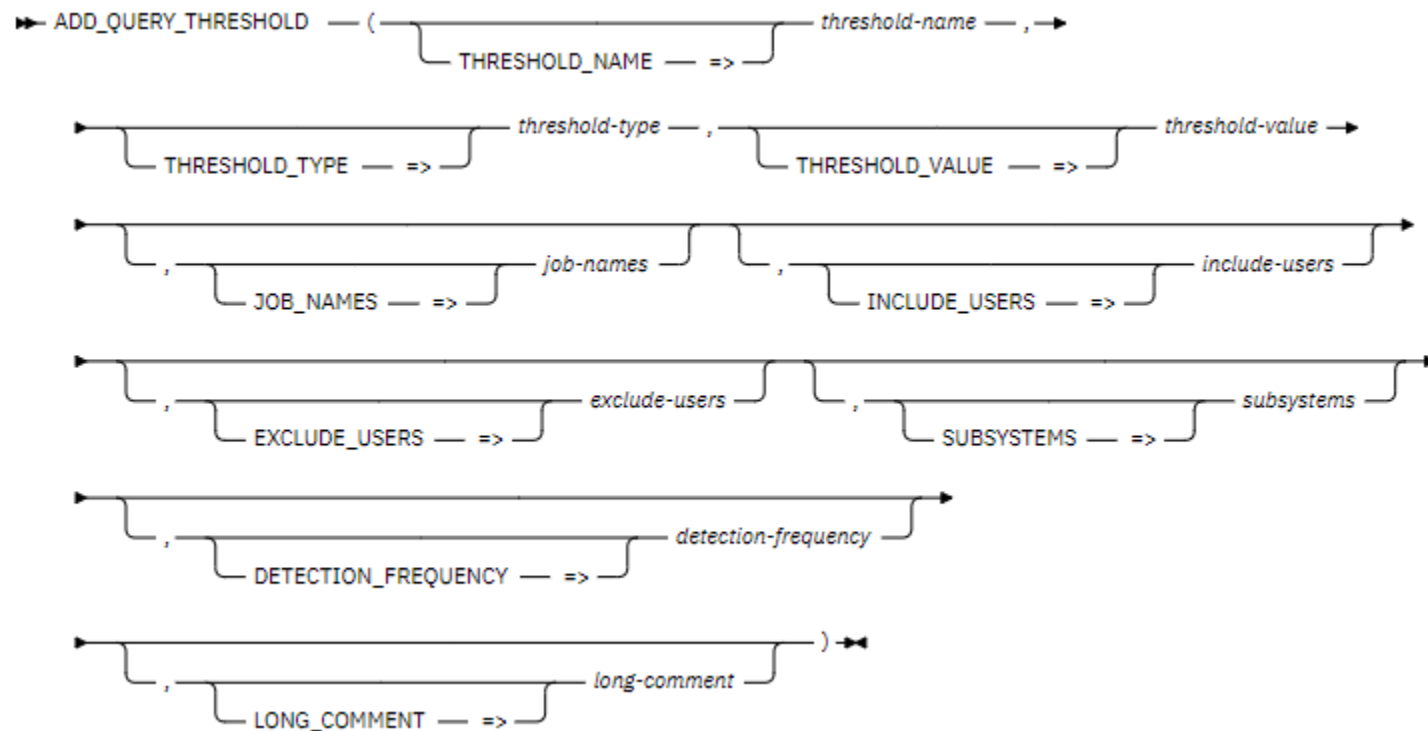
- Services IBM i

Services	Type	Description
ADD_QUERY_THRESHOLD	Procédure	Ajout d'un seuil pour le superviseur de requêtes
QUERY_SUPERVISOR	Vue	Règles et seuils du superviseur de requêtes
REMOVE_QUERY_THRESHOLD	Procédure	Suppression d'un seuil pour le superviseur de requêtes
END_IDLE_SQE_THREADS	Procédure	permet d'arrêter tous les threads du jobs qui ne sont pas utilisés (permet d'éviter CPF180B "Function XXXX is not allowed in a job which has multiple threads.")

- Pas de possibilité de modification
 - REMOVE + ADD

Query Supervisor

- Ajouter un seuil



Query Supervisor

■ Paramètres

- threshold-name : nom du seuil
- threshold-type : ressource à surveiller
 - CPU TIME
 - ELAPSED TIME
 - TEMPORARY STORAGE
 - TOTAL IO COUNT
- threshold-value : valeur du seuil (entier)
- include-users : utilisateurs à inclure (liste jusqu'à 100)
 - Défaut : *ALL
- exclude-users : utilisateurs à exclure (liste jusqu'à 100)
 - Défaut : *NONE
- job-names : liste de noms de travaux (100 maximum)
 - Noms génériques possibles : QPADEV*
 - Défaut : *ALL
- subsystems : liste de noms de sous-systèmes (100 maximum)
 - Noms génériques possibles : QINTER*
 - Défaut : *ALL
- detection-frequency : fréquence d'échantillonnage (entier en seconde)
 - Défaut = 600 (10 minutes)
- long-comment : commentaire

Query Supervisor

- Exemple
 - Ajout de seuils
 - Procédure QSYS2.ADD_QUERY_TRESHOLD

```
CALL QSYS2.ADD_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXTIME',
                              THRESHOLD_TYPE => 'CPU TIME',
                              THRESHOLD_VALUE => 6000,
                              LONG_COMMENT => 'Maximum runtime for all jobs');

CALL QSYS2.ADD_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXTIME QZDASOINIT NB',
                              THRESHOLD_TYPE => 'CPU TIME',
                              THRESHOLD_VALUE => 180,
                              INCLUDE_USERS => 'NB',
                              JOB_NAMES => 'QZDASOINIT QZDASSINIT',
                              LONG_COMMENT => 'Maximum runtime for jobs user NB');

CALL QSYS2.ADD_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXSTG INTERACT NB',
                              THRESHOLD_TYPE => 'TEMPORARY STORAGE',
                              THRESHOLD_VALUE => 10,
                              INCLUDE_USERS => 'NB',
                              JOB_NAMES => 'QPADEV*',
                              LONG_COMMENT => 'Maximum storage for interactive jobs user NB');
```

Query Supervisor

- Exemple
 - Voir les seuils définis
 - Vue QSYS2.QUERY_SUPERVISOR

```
25 SELECT *
26 FROM qsys2.query_supervisor;
```

THRESHOLD_NAME	THRESHOLD_TYPE	THRESHOLD_VALUE	JOB_NAMES	INCLUDE_USERS	EXCLUDE_USERS	SUBSYSTEMS	DETECTION_FREQUENCY	LONG_COMMENT
Seuil	CPU TIME	1	-	PLB	-	-	30	-
MAXTIME	CPU TIME	6000	-	-	-	-	600	Maximum runtime for all jobs
MAXTIME QZDASOINIT NB	CPU TIME	180	QZDASOINIT,QZDASSINIT	NB	-	-	600	Maximum runtime for jobs user NB
MAXSTG INTERACT NB	TEMPORARY STORAGE	10	QPADEV*	NB	-	-	600	Maximum storage for interactive jobs user NB

Query Supervisor

- Exemple
 - Supprimer 1 seuil
 - Procédure QSYS2.REMOVE_QUERY_TRESHOLD

```

▶ REMOVE_QUERY_THRESHOLD ( threshold-name ) ▶
                        |
                        | THRESHOLD_NAME =>

```

```

CALL QSYS2.REMOVE_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXTIME');
CALL QSYS2.REMOVE_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXTIME QZDASOINIT NB');
CALL QSYS2.REMOVE_QUERY_THRESHOLD(THRESHOLD_NAME => 'MAXSTG INTERACT NB');

```

Query Supervisor

- Déclenchement

- Lorsqu'un seuil est dépassé, cela déclenche le programme indiqué via le point d'exit QIBM_QQQ_QRY_SUPER
 - Cf https://www.ibm.com/docs/en/i/7.5?topic=ssw_ibm_i_75/apis/xqrysuper.html

- A vous de fournir le programme ou les programmes
 - Il faut respecter l'interface demandée
 - Cf format QRYS0100
 - Et quelques contraintes techniques
 - Doit être défini dans l'ASP systèmes
 - Doit être ACTGRP(*CALLER)
 - Doit être thread safe. Pour ILE RPG : THREAD(*CONCURRENT)
 - Pas d'accès à la BD (ni SQL ni accès natif)
 - Le programme d'exit s'exécute dans le travail : éviter d'utiliser le contrôle transactionnel

Query Supervisor

- Déclenchement

- IBM fournit des exemples
 - <https://www.ibm.com/docs/en/i/7.5?topic=supervisor-query-example-exit-programs>
- Cataloguer votre programme


```
ADDEXITPGM EXITPNT(QIBM_QQQ_QRY_SUPER) FORMAT(QRYS0100) PGMNBR(*LOW)
          PGM(<your library>/<your program>) THDSAFE(*YES)
          TEXT('Query Supervisor send QSYSOPR message')
```
- Que pouvez-vous faire ?
 - Remonter un message (*SYSOPR ou autre)
 - Arrêter la requête
 - Effectuer une capture du plan cache
 - Logger l'information
 - Remonter l'alerte dans un outil de supervision
 - ...

Query Supervisor

- Déclenchement
 - Règles d'écriture du programme d'exit
 - Ne doit pas planter !
 - Doit être rapide
 - Capturer l'événement et les informations
 - Faire les actions en asynchrone
 - Soumission
 - File de donnée
 - Message
 - ...

Query Supervisor

- Déclenchement
 - Si aucun programme n'est enregistré, la requête n'est pas arrêtée
 - Si un (ou plusieurs) programme est enregistré
 - Le programme décide de continuer ou arrêter la requête
 - Via un code retour transmis à DB2

Query Supervisor

- Déclenchement

- En cas d'erreur sur un programme d'exit
 - Un message CPD43B0 est envoyé à *SYSOPR (première occurrence d'erreur par jour)

```
Message ID . . . . . : CPD43B0          Severity . . . . . : 80
Message type . . . . . : Diagnostic
Date sent . . . . . : 03/22/21        Time sent . . . . . : 13:04:46

Message . . . . . : An exit point program registered to exit point
QIBM_00Q_QRY_SUPER encountered an error.
Cause . . . . . : An error was encountered while resolving to or running one
or more exit programs registered to exit point QIBM_00Q_QRY_SUPER. The first
exit program was SUPERTERM in library SCOTTF. The current user was TIMMR.
Refer to message ID MCH3401 in the job log for job 212791/TIMMR/QDFTJOB0D for
more information about the error.
```



Infrastructures du
futur

7 et 8 novembre 2023

Université **IBMi**

7 novembre 2023

3. QAQQINI



Let's
Create

Query Options File

- Ce fichier contient des attributs
 - Vous pouvez modifier les valeurs pour agir sur le comportement de vos requêtes

- Vous le trouverez
 - QSYS : fichier modèle, non pris en compte
 - QUSRSYS : fichier pris par défaut au niveau système
 - XXX : dans n'importe quelle bibliothèque

- Pour créer un fichier QAQQINI


```
CRTDUPOBJ OBJ(QAQQINI)
          FROMLIB(QSYS)
          OBJTYPE(*FILE)
          TOLIB(XXX)
          DATA(*YES)
          TRG(*YES)
```

- Référence
 - <https://www.ibm.com/docs/en/i/7.5?topic=attributes-qaqqini>

Query Options File

- Le fichier dispose de trigger
 - Ne pas copier par CPYF

- Il est composé de 3 zones
 - QQPARM : nom de l'attribut
 - QQVAL : la valeur
 - QQTEXT : description

- Vous pouvez ensuite modifier les différents attributs


```
UPDATE XXX/QAQQINI SET QQVAL='*YES'
WHERE QQPARM='MESSAGES_DEBUG'
```

Query Options File

```
11 select * from qusrsys.qaqqini ;|
```

QOPARM	QQVAL	QQTEXT
APPLY_REMOTE	*DEFAULT	Specifies for database queries involving distributed files, whether or not the CHGQRYA query attributes...
PARALLEL_DEGREE	*DEFAULT	Specifies the parallel processing option that can be used when running database queries and keyed acces...
ASYNC_JOB_USAGE	*DEFAULT	Specifies the circumstances in which asynchronous (temp writer) jobs can be used to help process databa...
QUERY_TIME_LIMIT	*DEFAULT	Specifies a time limit for database queries allowed to be started based on the estimated number of elap...
UDF_TIME_OUT	*DEFAULT	Specifies the amount of time, in seconds, that the database will wait for a User Defined Function (UDF)...
MESSAGES_DEBUG	*DEFAULT	Specifies whether query optimizer debug messages that would normally be issued if the job was in debug ...
PARAMETER_MARKER_CONVERSION	*DEFAULT	For dynamic SQL queries, specifies whether or not to allow literals to be implemented as parameter mark...
OPEN_CURSOR_THRESHOLD	*DEFAULT	Specifies the threshold to start full close of pseudo closed cursors. QQVAL: *DEFAULT--Is equivalent to...
OPEN_CURSOR_CLOSE_COUNT	*DEFAULT	Specifies the number of cursors to full close when threshold is encountered. QQVAL: *DEFAULT--Is equiva...
OPTIMIZE_STATISTIC_LIMITATION	*DEFAULT	Specifies limitations on query optimizer's statistics gathering. QQVAL: *DEFAULT--The amount of time sp...
OPTIMIZATION_GOAL	*DEFAULT	Specifies the goal that the query optimizer should use when making costing decisions. QQVAL: *DEFAULT--...
FORCE_JOIN_ORDER	*DEFAULT	Specifies that the join of tables is to occur in the order specified in the query. QQVAL: *DEFAULT--The...
COMMITMENT_CONTROL_LOCK_LIMIT	*DEFAULT	Specifies the maximum number of records which can be locked to a commit transaction initiated after set...
REOPTIMIZE_ACCESS_PLAN	*DEFAULT	For queries with a saved access plan, this option specifies to the query optimizer to reoptimize the qu...
SQLSTANDARDS_MIXED_CONSTANT	*DEFAULT	For SQL queries, this parameter specifies whether or not to allow IGC constants to always be treated as...
SYSTEM_SQL_STATEMENT_CACHE	*DEFAULT	Specifies for dynamic SQL queries that are not stored in an SQL package the ability to disable system w...

Query Options File

- Aucun droit nécessaire à l'exécution
 - Ce ne sont pas les droits de l'utilisateur qui comptent
 - Le fichier est pris en compte par l'optimiseur qui est un process système

- Conseil
 - Limiter les droits à *PUBLIC *USE
 - Permet à chacun de voir le contenu
 - *CHANGE uniquement pour l'administrateur / administrateur de la base de donnée
 - Le changement de comportement peut impacter le système dans sa globalité

Query Options File

- En cas de changement de version / TR
 - Nouvelles options
 - Modifications de valeurs par défaut

- Pensez à mettre à jour vos fichiers QAQQINI


```
insert into nb.qaqqini
select * from qsys.qaqqini b
where qqparm not in (select qqparm from nb.qaqqini)
```

Query Options File

- Pour prendre en compte un fichier QAQQINI pour le travail en cours

CHGQRYA QRYOPTLIB(NB)

- Ou

OVRDBF QAQQINI TOFILE(XXX/QAQQINI) SCOPE(*JOB)

- Intérêt

- Régler un comportement global pour le système
- Et des comportements « exotiques »

- Par exemple pour des usages de type infocentre

MEMORY_POOL_PREFERENCE (*DEFAULT -> *BASE ou nn)

OPTIMIZATION_GOAL (*DEFAULT -> *FIRSTIO)

Query Options File

- Trouver tous les fichiers QAQQINI

```
SELECT OBJNAME, OBJLIB, X.LAST_USED_TIMESTAMP
FROM TABLE (QSYS2.OBJECT_STATISTICS('*ALL', 'FILE', 'QAQQINI')) X
```

- Les valeurs modifiées

```
SELECT *
FROM QAQQINI WHERE QQVAL <> '*DEFAULT'
```

Query Options File

- Depuis la V6R1, nous avons une procédure
 - OVERRIDE_QAQQINI
 - Permet de dynamiquement modifier un attribut dans un travail
 - Ne nécessite pas de créer un fichier QAQQINI spécifique et permanent
 - Il est créé dans QTEMP

- Exemple

```
-- Création d'un fichier QAQQINI dans QTEMP
```

```
CALL override_qaqqini('1' , ' ' , ' ');
```

```
-- Modification des valeurs le job ici pour utiliser les MQTs
```

```
CALL override_qaqqini('2' , 'MATERIALIZED_QUERY_TABLE_REFRESH_AGE' , '*ANY');
```

```
CALL override_qaqqini('2' , 'MATERIALIZED_QUERY_ TABLE_USAGE' , '*ALL');
```

```
-- Suppression de qaqqini de QTEMP, si nécessaire
```

```
CALL override_qaqqini('3' , ' ' , ' ');
```

Query Options File

- Aucun droit n'est nécessaire, sauf pour les options suivantes
 - Nécessitent *JOBCTL
 - Ou être autorisé à la fonction d'usage QIBM_DB_SQLADM
 - QUERY_TIME_LIMIT si different de 0
 - STORAGE_LIMIT
 - PARALLEL_DEGREE si different de *NONE
 - PARALLEL_MAX_SYSTEM_CPU

Exemple

- En général on va trouver au moins 2 QAQQINI
 - QAQQINIBCH – Pour les batchs (plus exactement tous les travaux *ALLIO)
 - QAQQINIINT – Pour les travaux de type interactif (travaux plutôt *FIRSTIO)

- Plus de réglages ?
 - Vous utilisez les MQTs ?
 - Vous avez de nombreux processeurs ?
 - Vous avez des pools de mémoires partagés spécifiques ?
 - ...

Exemple

- Les valeurs critiques
 - OPTIMIZATION_GOAL
 - *DEFAULT : utilise le réglage de l'interface
 - ODBC, options de pré-compilation SQL, clause OPTIMIZE FOR nnn ROWS
 - *FIRSTIO – Optimize for quickest retrieval of first few rows
 - *ALLIO – Optimize for quickest retrieval of all the rows

Exemple

- Les valeurs critiques
 - QUERY_TIME_LIMIT (en secondes)
 - Limite autorisée avant démarrage de l'exécution, basée sur l'estimation de SQE
 - *DEFAULT : utilise la valeur système QQRYTIMLMT
 - Nous vous recommandons une valeur de 12h maximum (43.200)
 - OK pour la plupart des requêtes
 - Si estimation > 12h : est-il pertinent de déclencher ces requêtes ? Il faudra travailler les index ...
 - Certains usages peuvent nécessiter plus : infocentre ? Dans ce cas, prévoir un QAQQINI spécifique
 - Vous pouvez aussi utiliser Query Supervisor

Exemple

- Les valeurs critiques
 - STORAGE_LIMIT (en Mo)
 - Stockage temporaire maximum autorisé (estimation)

 - *DEFAULT : *NOMAX

 - Par sécurité : indiquer un pourcentage de l'espace libre
 - Permet de se prémunir contre une requête « incontrôlée »

Exemple

- Les valeurs critiques
 - MEMORY_POOL_PREFERENCE
 - *DEFAULT – Pool mémoire du travail
 - *JOB – Pool mémoire du travail
 - *BASE – Utilisation du pool *BASE
 - *NAME poolname – utiliser un pool mémoire nommé
 - *PRIVATE lib/SBS/PoolNumber – Utiliser un pool mémoire privé
 - nn – Utiliser le pool mémoire nn

Exemple

- Les valeurs critiques
 - MATERIALIZED_QUERY_TABLE_USAGE (MQT)
 - *DEFAULT – Idem*NONE
 - *NONE – Ne pas utiliser les MQTs
 - *ALL – Les MQTs maintenus par l'utilisateur peuvent être utilisées
 - *USER – Les MQTs maintenus par l'utilisateur peuvent être utilisées
 - MATERIALIZED_QUERY_TABLE_REFRESH_AGE
 - Indique l'âge de validité (depuis le dernier refresh) d'un MQT
 - *DEFAULT – 0
 - 0 – Aucun MQT ne peut être utilisé
 - *ANY – Aucun contrôle de l'âge
 - Timestamp_duration – un horodatage

Exemple

- Supprimée en 7.4
 - SQE_NATIVE_ACCESS *NO
 - Permettait de favoriser CQE plutôt que SQE
 - Utilisé en 7.2 et 7.3

SMP (Symetric MultiProcessing)

- Option 26 de SS1, désormais gratuite !
 - Nécessite plusieurs processeurs

- Permet de répartir l'exécution d'une requête sur plusieurs processeurs
 - Très efficace sur les grosses requêtes
 - Potentiellement dangereux également

- Peut être activé au niveau système ou travail
 - Ne présage pas de l'usage par l'optimiseur

- L'utilisation des ressources peut être « agressive »
 - IBM recommande de limiter aux requêtes les plus longues
 - IBM recommande un niveau de traitement parallèle à *OPTIMIZE

SMP (Symetric MultiProcessing)

- Usages
 - SQL, OPNQRYP, Query/400
 - Création d'index, mise à jour d'index
 - RGZPFM
 - Commandes type CPYFRMIMPF
 - Db2 Mirror
 - ...

- Pour activer SMP
 - Système ou travail : QAAQINI PARALLEL_DEGREE ou CHGQRYA DEGREE()
 - Travail : Registre spécial CURRENT DEGREE
 - Système : Valeur système QQRYDEGREE
 - Déconseillé

SMP (Symetric MultiProcessing)

- Valeurs possibles de PARALLEL_DEGREE
 - *DEFAULT – utilise la valeur système QQRYDEGREE
 - *OPTIMIZE – détermine la meilleure consommation de ressources (IO et tâches)
 - *IO – n'utilise pas SMP
 - *OPTIMIZE nnn (de 1 – 200) %
 - *MAX – Utilise toutes les ressources estimées pour la requête
 - *MAX nnn (de 1 -200) %
 - *NUMBER_OF_TASKS nnn – nombre maximum de tâches par requête. IBM le déconseille avec SQE

- D'autres valeurs entrent en compte, principalement si *OPTIMIZE est indiqué
 - PARALLEL_MAX_SYSTEM_CPU
 - PARALLEL_MIN_TIME



Infrastructures du
futur

7 et 8 novembre 2023

Université **IBM i**

7 novembre 2023

Q/R



Let's
Create

