Une introduction à la gestion de la performance

Paul Koufalis
Président
Progresswiz Informatique

Le groupe d'utilisateurs Progress du Québec

Progresswiz

Progresswiz Informatique

- **Offre de l'expertise technique Progress, UNIX, Windows et plus depuis 1999**
- **Spécialisé en matière de performance, disponibilité des systèmes et planification de la continuité d'affaires**

Agenda

- **Introduction**
- **Les « vrais » paramètres par défaut**
- **Promon**
- **Iostat, vmstat, nmon et autres outils UNIX**
- **Les liens entre le O.S. et la BD**
- **Questions**

Avant de commencer

- **Désolé – pas de Windows aujourd'hui**
  - **Les principes sont les mêmes qu'en UNIX**
- **Si vous avez des questions, n'hésitez surtout pas!**
  - **Par contre, on va laisser les grosses questions existentielles pour la fin**

Avant de commencer

- **Cette présentation n'est qu'un survol de la gestion de la performance**
  - **On ne peut pas compresser deux semaines de formation dans une présentation d'une heure!**
- **Et finalement: *Your Mileage May Vary***
  - **L'information dans cette présentation ne s'applique pas nécessairement à votre environnement**
  - **N'hésitez pas de demander une évaluation ou de la formation de votre DBA Progress préféré**

Introduction

- **C’est quoi exactement *la gestion de la performance*?**
  - **C’est quoi le problème? Il faut identifier un problème avant de trouver une solution**
- **Il faut bien savoir comment identifier et documenter un problème précis**

Introduction

- « C'est lent »
  - Hmmm…non
- Le rapport 32.12 prenait 15 minutes et maintenant prend 2 hrs
  - Mieux
- Un problème doit être quantifiable
  - Sinon, comment est-ce que je peux mesurer si ma solution a résolu le problème?

Le début...

- **Du livre « Progress Performance Tuning Guide » de Dan Foreman**
  - **www.bravepoint.com**
  - **Le livre sur la performance Progress**
- **En générale, je commence avec ces paramètres**
  - **Ça devient le nouveau « baseline » pour les améliorations futures**

Les vrais params par défaut

- **-bibufs: 25-50**
- **-bi (cluster size): 1Mb – 32Mb**
- **-blocksize (BD): 8K**
- **-biblocksize: 8K ou 16K**
- **-B: à discuter mais généralement le plus possible**
- **-directio: à discuter mais typiquement sur AIX seulement**
- **-spin: 10 000 X # CPU**
- **-semsets: 1 par 100 utilisateurs**
- **APW: 2-4**
- **AIW/BIW: Oui (vous avez activé le AI, non?)**

Les outils

- **Progress:**
  - **Promon**
  - **VST**
  - **DB Analysis**
- **UNIX:**
  - **iostat**
  - **vmstat**
  - **sar**
  - **Nmon/glance/topas/etc**

Promon

- **Connaissez-vous le menu caché R&D?**

```
PROGRESS MONITOR Version 9

Database: /database/sports

1.  User Control
2.  Locking and Waiting Statistics
3.  Block Access
4.  Record Locking Table
5.  Activity
6.  Shared Resources
7.  Database Status
8.  Shut Down Database

T.  Transactions Control
L.  Resolve Limbo Transactions
C.  Coordinator Information

M.  Modify Defaults
Q.  Quit

Enter your selection: R&D
```


Promon

```
12/12/05        Progress Version 9 Monitor (R&D)
18:57:58        Main (Top) Menu

                 1. Status Displays ...
                 2. Activity Displays ...
                 3. Other Displays ...
                 4. Administrative Functions ...
                 5. Adjust Monitor Options

Enter a number, <return>, P, T, or X (? for help):
```


Promon

```
12/12/05        Progress Version 9 Monitor (R&D)
18:59:39        Activity Displays Menu

                 1. Summary
                 2. Servers
                 3. Buffer Cache
                 4. Page Writers
                 5. BI Log
                 6. AI Log
                 7. Lock Table
                 8. I/O Operations by Type
                 9. I/O Operations by File
                10. Space Allocation
                11. Index
                12. Record
                13. Other

Enter a number, <return>, P, T, or X (? for help):
```


Promon

```
12/12/05        Activity: Summary
19:00:38        12/10/05 07:53 to 12/12/05 18:57 (59 hrs 4 min)

Event                  Total  Per Sec |Event                  Total  Per Sec

Commits              9146537     43.0 |DB Reads            12644563     59.4
Undos                    181      0.0 |DB Writes           10683978     50.2
Record Reads      4134260302   1264.0 |BI Reads              219602      1.0
Record Updates       8983228     42.2 |BI Writes            2560633     12.0
Record Creates       8528162     40.1 |AI Writes            1133788      5.3
Record Deletes       4125435     19.4 |Checkpoints             530      0.0
Record Locks       300922580   1415.1 |Flushed at chkpt       12127      0.0
Record Waits            6048      0.0

Rec Lock Waits      0 %    BI Buf Waits      0 %    AI Buf Waits      0 %
Writes by APW      99 %    Writes by BIW    74 %    Writes by AIW    95 %
DB Size:           29 GB   BI Size:       1376 MB   AI Size:       1166 MB
Empty blocks:1205609       Free blocks:   6218      RM chain:    104314
Buffer Hits         0 %    Active trans:     3

10 Servers, 39 Users (36 Local, 3 Remote, 36 Batch), 6 Apws

Enter <return>, A, L, R, S, U, Z, P, T, or X (? for help):
```


Summary

- **Attention aux chiffres bizarres**
  - **0% de buffer hits?**
- **Les compteurs sont 32 bits**
  - **Quand ils arrivent au max ils retournent à zéro**

Promon

```
12/12/05        Activity: Buffer Cache
19:01:59        12/10/05 07:53 to 12/12/05 18:57 (59 hrs 4 min)

                                Total   Per Min   Per Sec    Per Tx

Logical reads                4111733841   1160160  19336.00      3.44
Logical writes                212186756     59820    997.00      4.41
O/S reads                      12644989      3568     59.46      1.38
O/S writes                     10683978      3014     50.24      1.16
Checkpoints                         530         0      0.00      0.00
Marked to checkpoint           9977065      2815     46.91      1.09
Flushed at checkpoint            12127         3      0.05      0.00
Writes deferred               201515704     56820    947.00      3.24
LRU skips                       2060278       581      9.68      0.22
LRU writes                         1949         0      0.00      0.00
APW enqueues                     408327       115      1.92      0.04

Hit Ratio: 0 %

Enter <return>, A, L, R, S, U, Z, P, T, or X (? for help):
```

Real Buffer Hits = $\frac{(4\,111\,733\,841 + 212\,186\,756) - (12\,644\,989 + 10\,683\,798)}{(4\,111\,733\,841 + 212\,186\,756)}$

= 99.46%

Buffer Hits

- **La « norme » semble être 95% ou plus**
  - **Non – 99% ou plus**
- **Pourquoi?**
  - **95% = 95 lectures sur 100 sont satisfait en mémoire et 5 sur disque**
  - **99% = Une lecture sur 100 sur disque**
    - **99% est 5 FOIS MEILLEUR que 95%**
    - **99.9% = 1:1000 = 10 X meilleur que 99%**

Buffer Hits

- **Comment atteindre 99%**
  - **Augmenter le –B**
    - **Ça va peut-être nécessiter la migration vers une plate-forme 64bit**
  - **Corriger les programmes qui font trop de lectures pour rien**
    - **Essayer des outils comme le profiler de Progress**
      - **Gratuit mais non-supporté**

Promon

```
12/12/05        Activity: BI Log
19:02:49        12/10/05 07:53 to 12/12/05 18:57 (59 hrs 4 min)

                                Total   Per Min   Per Sec    Per Tx

Total BI writes               2560633       722     12.04      0.27
BIW BI writes                 1916793       541      9.01      0.20
Records written             241722218     68160   1136.00      2.94
Bytes written               591382756    166860   2781.00      3.61
Total BI Reads                 219602        62      1.03      0.02
Records read                    52272        14      0.24      0.00
Bytes read                    3489027       984     16.40      0.38
Clusters closed                   530         0      0.00      0.00
Busy buffer waits              570595       161      2.68      0.06
Empty buffer waits                  0         0      0.00      0.00
Log force waits                     0         0      0.00      0.00
Log force writes                    0         0      0.00      0.00
Partial writes                 384877       108      1.80      0.04

Enter <return>, A, L, R, S, U, Z, P, T, or X (? for help):
```

BI Waits

- **Rien de pire que de faire attendre une écriture au BI**
  - **Éliminer les « Empty Buffer Waits »**
  - **Il va toujours avoir un peu de « Busy Buffer Waits »**
- **Les « Partial Writes » sont normales**
  - **C'est le –Mf qui force la BD à écrire les changements avant que le BI buffer soit plein**

Promon

```
12/12/05        Activity: Other
19:03:51        12/10/05 07:53 to 12/12/05 18:57 (59 hrs 4 min)

                                Total   Per Min   Per Sec    Per Tx

Commit                        9146537      2581     43.01      1.00
Undo                              181         0      0.00      0.00
Wait on semaphore             1656943       467      7.79      0.18
Flush master block                662         0      0.00      0.00
```

Wait on Semaphore

- **Pas de chiffre magique mais en bas de 10/seconde est acceptable**
- **Si plus haut il suffit souvent d'augmenter le –spin (Enterprise DB)**
- **Si –spin est déjà assez haut il y a une autre contention à quelque part**

Promon

```
12/12/05        Activity: Performance Indicators
19:04:32        12/10/05 07:53 to 12/12/05 18:57 (59 hrs 4 min)

                                Total   Per Min   Per Sec    Per Tx

Commits                       9146537      2581     43.01      1.00
Undos                             181         0      0.00      0.00
Index operations            171197447     48300    805.00      4.62
Record operations          4155897127   1172580  19543.00      3.57
Total o/s i/o                27246784      7680    128.00      2.97
Total o/s reads              12867588      3631     60.51      1.40
Total o/s writes             14379196      4057     67.62      1.57
Background o/s writes        14360141      4052     67.53      1.57
Partial log writes            1518475       428      7.14      0.16
Database extends                    0         0      0.00      0.00
Total waits                   1657559       467      7.79      0.18
Lock waits                       6354         1      0.02      0.00
Resource waits                1651205       466      7.76      0.18
Latch timeouts                2989279       843     14.05      0.32

Buffer pool hit rate:   0 %

Enter <return>, A, L, R, S, U, Z, P, T, or X (? for help):
```


Indicateurs de performance

- **Il n'y a pas de chiffre magique ici**
- **Essayez de garder les « waits » aussi bas que possible**
- **Souvent tous les « wait » viennent du « Wait on semaphore »**

Promon

```
12/12/05        Checkpoints
19:04:56

Ckpt                          ------ Database Writes ------
 No. Time       Len    Dirty   CPT Q    Scan   APW Q Flushes

 537 19:03:42     0    12186    6933     522       0       0

 536 19:02:15    87    10473   10554     418       0       0
 535 19:00:56    79    13078   13159     330       0       0
 534 18:59:09   107    10658   10666     623       0       0
 533 18:58:01    68     9088    9184     250       0       0
 532 18:56:54    67    16549   16629     257       0       0
 531 18:54:56   118    17178   17139     708       0       0
 530 18:52:41   135    18056   18022     854       0       0

Enter <return>, R, P, T, or X (? for help):
```


Checkpoints

- **Progress utilise des « fuzzy checkpoint »**
  - **Demandez moi de vous l'expliquer après la présentation**
  - **De façon simple, tous les changements à la BD doivent être écrit au disque à la fin du CKPT**
  - **Sinon, la BD arrête tous les TX le temps de finir**
    - **Les blocs qui restent à écrire sont les « Buffers flushed at CKPT »**
    - **À éviter!**

Longueur des CKPT

- **Minimum une minute quand il y a grand volume de transaction**
  - **Si moins d'une minute les APW's n'auront probablement pas le temps de finir leur travail**
- **Typiquement aux 15 minutes**
- **Le temps de CKPT est inversement proportionnel à la taille du cluster BI**

Index Analysis

- proutil <nom-bd> -C idxanalys
- Regarder les colonnes « % Util » et « Level »
  - Garder le % d'utilisation aussi haut que possible
  - En bas de 80% il faut penser à des idxbuild ou idxcompact
    - Attention au idxcompact avant 9.1E
  - L'idxbuild réduira aussi le nombre de niveau
  - Le plus de niveau, le plus de lecture de blocs d'index pour éventuellement trouver le rowid de l'enregistrement voulu
- Ignorer les index de moins d'un Mg

Table Analysis

- proutil <nom-bd> -C tabanalys
- Je ne l'utilise pas vraiment
- Il y a du monde qui regarde le « scatter factor » mais c'est relatif
  - Dans quel ordre est-ce que vous accéder vos données le plus souvent?
    - Index principal?
- Essayer de faire des dump and load sur une de vos tables volumineuse
  - Utiliser des index différent pour chaque dump et essayer le même FOR EACH après chaque load
    - Il va avoir une différence assez importante

_Tablestat et _indexstat

- **Des VST qui vous donnent des statistiques d'utilisations de la table ou l'index**
- **Utiliser les paramètres de démarrage – tablebase, -tablelimit, -indexbase et – indexlimit pour activer le collecte de données pour toutes les tables et indexes**

_Tablestat et _indexstat

```
File-Name: customer
   TableStat-id: 1
           read: 1998806512
         update: 74157
         create: 28656
         delete: 1548

Index-Name: name
   IndexStat-id: 15
           read: 2021345680
         create: 59476
         delete: 17291
          split: 447
    blockdelete: 3
```

Le serveur

- **Trois sous-systèmes principaux:**
  - **CPU**
  - **Mémoire**
  - **Disque**
- **Ils sont tous facile à surveiller**
  - **Par contre, c'est difficile des fois de trouver le processus qui en abuse**

CPU

- L'utilisation CPU est séparé en 4 catégories:
  - Utilisateur, système et IO wait et idle
- Utilisateur: Le vrai travail
  - On veut maximiser le temps CPU de l'utilisateur
- Système: Tous le travail que le système d'exploitation doit faire en arrière-plan pour supporter les processus
- Idle: Du temps ou le CPU ne fait rien
- IO wait: Du temps ou le CPU ne fait rien MAIS il y a au moins un processus qui attend après du IO

CPU

- **Le wait IO n'est pas nécessairement mauvais**
  - **Si le système n'a rien à faire il va accorder plus de temps CPU à chaque processus**
  - **Une application BD fait normalement beaucoup d'IO, alors s'il est accordé « trop » de temps CPU il va attendre**
    - **Résultat: Wait IO**
- **S'il y a beaucoup de wait IO dans un système occupé (idle à zéro) il se peut qu'il aie un bottleneck aux disques**

CPU

```
# sar 5 5
20:34:48     %usr     %sys     %wio   %idle
20:34:53       33       14       17      36
20:34:58       37       17       12      34
20:35:03       37       15       17      32
20:35:08       33       27       16      24
20:35:13       48       12       13      27

Average        38       17       15      31
```


Mémoire

- **Il ne devrait jamais avoir de la mémoire libre**
- **Si un processus ne l'utilise pas, le système d'exploitation devrait l'utiliser pour ces tampons fichiers**
- **ATTENTION: Pas de pagination vers la mémoire virtuelle**
  - **Il n'y a rien de pire pour tuer la performance!**

Mémoire

```
# vmstat 5 5
kthr     memory             page              faults        cpu
----- ----------- ------------------------ ------------ -----------
 r  b   avm   fre    re  pi  po  fr   sr  cy  in   sy    cs   us sy id wa
 6  3 1474165  1213   0   0   0  31   38   0 225 102734 15087 29 14 45 13
 5  2 1473857  1919   0   0   0 886 2049   0 5988 77728 21176 29 13 35 22
 2  3 1474371  1418   0   0   0 787 1932   0 6075 74648 16959 27 11 31 31
 2  3 1474360  1119   0   0   0 565 1470   0 6195 67909 18500 27 12 38 24
10  2 1471987  7287   0   0   0 727 1733   0 5805 96440 17757 47 18 22 13
```


Les disques

- **Toujours un point d'arguments, ces disques**
- **Les vendeurs de carrosseries essaient de nous vendre du RAID 5 (ou leur propre SuperHyperNotRaid5RAID)**
  - **Oh well…c'est normale…ils ont une vente à faire et ils peuvent offrir le même espace disque effectif avec moins d'espace brut**

Les disques

- **Normalement, on veut le BI sur un disque à part des .d***
  - **Le BI ne profite pas de systèmes de fichiers *striped***
  - **On ne veut surtout pas ralentir les écritures au BI**
  - **Et pendant qu'on en parle, arrêter de tronquer votre BI à tous les soirs**

La vérité et toute la vérité sur RAID5

- **RAID 5 is NOT EVIL**
  - **Sans rentrer dans une explication détaillée, on peut dire que le RAID 5 fait plus d'opérations d'écritures que le RAID 10 (stripe and mirror)**
  - **Pour atténuer l'effet, les manufacturiers mettent des tampons de mémoire entre le serveur et le disque**
  - **Sur des systèmes bas de gamme, les tampons sont directement sur la carte SCSI/RAID**
  - **Si les tampons se remplissent, le sous-système disque agit effectivement comme s'il n'avait aucun tampon**

RAID 5

- **Pensez à un évier**
  - **Le robinet = demandes d'écritures du serveur**
  - **Le drain = les écritures actuelles aux disques**
  - **Si l'évier se remplit, il faut arrêter le robinet jusqu'à temps que le drain puisse faire de l'espace**
- **Dans des gros SAN, l'évier est tellement gros que ça prend un méchant robinet pour le remplir!**

Le risque du RAID 5

- **Dans des systèmes bas de gamme, il y a un vrai risque de corruption de BD si le serveur plante**
  - **Encore une fois un sujet qui prendrait trop de temps à expliquer**
- **Pour éliminer le risque, il faut arrêter l'utilisation des tampons lors des écritures**
  - **Bye-bye à la performance**
- **Encore, dans un vrai SAN, toutes les composantes sont redondant et protéger par des systèmes de UPS avancés**

Le mot final sur le RAID 5

- **Il y aura un prix de performance à payer**
  - **Mais on ne sait pas à quel débit et si ce débit est suffisant pour votre environnement**
- **Si ce n'est pas un vrai SAN, éviter le RAID 5**
  - **La différence de coût est négligeable comparer au coût si jamais il y a un problème**

La surveillance des disques

- **Les Kb/sec ne m'intéressent pas**
  - **C'est les opérations par secondes qui sont importantes**
- **Un disque normale peu faire environ 100 IOPS**
  - **150 pour les disques 15 000 tours**
  - **Souvent encore plus pour les disques SAN**

La surveillance des disques

```
# iostat 5 5
Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk39          1.6      26.2       3.8         84        48
hdisk40          4.4     143.9      26.2        292       432
hdisk41          3.6     177.3      15.7        220       672

# sar –d 5 5
21:16:18     device    %busy    avque    r+w/s   blks/s   avwait   avserv
             hdisk39      2      0.2        6       50      0.0      0.0
             hdisk40      3      0.0       12       69      0.0      0.0
             hdisk41      2      0.0        8       57      0.0      0.0
```


La surveillance des disques

- **On veut valider quelques statistiques**
  - **Les tps sont plus ou moins égales pour toutes les disques des BDs**
  - **Avque et avwait sont zéro**
  - **Attention – le % activité des fois n'est pas exacte parce que les disques sont cachés en arrière du contrôleur RAID**

Nmon/Topas/Glance

- **Il existe plusieurs utilitaires de surveillance en temps réel**
- **Moi j'aime bien nmon mais ils sont tous pas mal équivalents**

nmon

```
CPU Utilisation                +--------------------------------------------------+
CPU    User%  Sys% Wait% Idle|0          |25         |50          |75       100|
 0      0.0   3.0   0.0  97.0|s                >                                 |
 1      2.5   4.0   0.0  93.6|Us                >                                |
 2     12.4  12.9  45.0  29.7|UUUUUUssssssWWWWWWWWWWWWWWWWWWWWW  >               |
 3     11.9   8.5  43.3  36.3|UUUUUssssWWWWWWWWWWWWWWWWWWWW   >                 |
 4     21.9  10.9   8.5  58.7|UUUUUUUUUUssssWWWW                    >           |
 5     28.4  12.4  25.9  33.3|UUUUUUUUUUUUUUssssssWWWWWWWWWWW                  >|
 6     31.8   7.0   0.5  60.7|UUUUUUUUUUUUUUUsss                    >           |
 7     13.9   3.0   0.0  83.1|UUUUUUs                               >           |
                               +--------------------------------------------------+
       15.4   7.7  15.4  61.6|UUUUUUUsssWWWWWW           >                      |
                               +--------------------------------------------------+
Memory Use  Physical    Virtual    Paging pages/sec  In     Out  VM parameters
% Used       100.0%       0.3%      to Paging Space   0.0    0.0 numperm  61.4%
% Free         0.0%      99.7%      to File System  504.3  375.1 minperm   4.9%
MB Used    14331.6MB     27.0MB    Page Scans      3621.8        maxperm   9.7%
MB Free        4.4MB   8165.0MB    Page Cycles        0.0        minfree 960
Total(MB)  14336.0MB   8192.0MB    Page Reclaim       0.0        maxfree 1216
Top Processes  Procs=1066 mode=3 (1=Basic, 2=CPU 3=Perf 4=Size 5=I/O w=wait-procs)
  PID    %CPU  Size   Res   Res   Res  Char RAM      Paging       Command
         Used    KB   Set  Text  Data   I/O Use io other repage
 809324  65.1  8196  7792  4036  3756     0  0%   62    5    1 _progres
 555136  47.2  7736  7332  4036  3296     0  0%   12  487    0 _progres
 527252  10.9  7272  6868  4036  2832     0  0%   36    0    0 _progres
   2838   2.5    20    20     0    20     0  0%    0    0    0 lrud
 799198   2.5  2064  1632   688   944 261242  0%    0    0    0 _mprosrv
 827024   2.0 1183512 1183116    16 1183100     0  8%    0    0    0 java
```


Questions?

Merci!