An Introduction to Performance Tuning

Paul Koufalis
President
Progresswiz Consulting

Progresswiz

Progresswiz Consulting

- **Based in Montréal, Québec, Canada**
- **Providing technical consulting in Progress®, UNIX, Windows, MFG/PRO and more**
- **Specialized in**
  - **Migrations**
  - **Performance tuning**
  - **Business continuity planning**
  - **Security**

Le groupe d'utilisateurs Progress du Québec

Progresswiz

Agenda

- **Introduction**
- **The "real" default parameters**
- **Promon**
- **Iostat, vmstat, nmon and other UNIX tools**
- **Questions**

Before We Start…

- **Sorry – no Windows today**
  - **The principals are the same as UNIX and apply directly**
- **If you have any questions please don't hesitate to stop me and ask**
  - **Except for "Is UNIX better than Windows?" (Hint: The answer is "it depends")**

Before We Start…

- **This presentation is a fly-by of performance tuning**
  - **We cannot cover a lifetime of knowledge and experience in one hour – sorry!**
- **And of course: *Your Mileage May Vary***
  - **The information presented here may or may not apply to your environment**
  - **It may sound self-serving but it's true: seek professional help**

Introduction

- **What exactly *is* performance tuning?**
- **You have to learn how to precisely identify and document a specific problem**

Introduction

- **“The system is slow”**
  - **Ummm…no**
- **Report 32.12 used to take 5 minutes, now it takes 43 minutes**
  - **Better**
- **A problem needs to be quantifiable**
  - **Otherwise, how do you know if you improved the situation?**

The Beginning

- **Before attacking specific problems, bring system to a baseline**
- **Many problems magically solved by setting some simple default parameters**
- **YMMV – Don’t go change all your params tomorrow!**

The Beginning…

- **Inspired by the "Progress Performance Tuning Guide" by Dan Foreman**
  - **www.bravepoint.com**
  - **BTW – this is the performance tuning bible**
    - **(Ask Scott for a PUG discount)**
- **Usually, I start with these parameters with some minor variations**

“Real” Default Startup Parameters

- -bibufs: 25-50
- -bi (cluster size): 2Mb – 8Mb
- -blocksize (DB): 8K (4K on Linux/Windows)
- -aiblocksize/-biblocksize: 16K
- -B: depends but generally as big as possible for small systems
- -directio: usually only on AIX
- -spin: 10 000 X # CPU
- -semsets: 1 per 100 users
- APW: 1 (add more only if you can prove why)
- AIW/BIW/WDOG: Yes (You ARE using AI, right?)

Tools

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

- **In V9 the R&D menu is hidden**

```
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
```

- **OpenEdge 10**

```
OpenEdge MONITOR Release 10

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

R&D.  Advanced options
  T.  2PC Transactions Control
  L.  Resolve 2PC Limbo Transactions
  C.  2PC Coordinator Information

  J.  Resolve JTA Transactions

  M.  Modify Defaults
  Q.  Quit

  Enter your selection:
```

Progresswiz

Promon

```
01/18/10        OpenEdge Release 10 Monitor (R&D)
10:45:04        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        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), 2 Apws

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


Activity – Summary

- **Watch out for weird numbers in V9**
  - **0% buffer hits?**
  - **32 bit counters roll over**

Activity – Buffer Cache

```
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

- People seem to accept 95% as good
  - Written in old PKB article
- No – 99% or more
- Why?
  - 95% = 5 IOs of 100 went to disk
  - 99% = 1 IO of 100 went to disk
    - 99% is 500% better than 95%, not 5%
    - 99.9% = 1:1000 = 1000% better than 99%

99% Buffer Hits

- **Increase –B**
  - **May require a migration to 64 bit**
- **Correct programming errors**
  - **Progress Profiler: free but unsupported**
  - **TableStat and IndexStat by User VST**

99% Buffer Hits

- **Many PPT's available online**
  - **"Pick an Index" – Mike Lonski**
    - **Get the demo code**
  - **"Improving your Application Performance" – Alan Webb**
  - **"Super Efficient Indexing" – Dan Foreman**

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

- Very few things worse than making a BI write wait
  - Eliminate “Empty Buffer Waits”
    - Increase -bibufs
  - There will always be some “Busy Buffer Waits”
- “Partial Writes”are normal
  - – Mf forces BI buffer writes to disk even if buffer not full

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

- **Less than 10/sec ok**
- **First fix is to increase –spin**

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):
```

Performance Indicators

- Keep an eye on “Database extends” if using variable extents
- Try to keep any “waits” as low as possible
- Often, all the “Resource waits” are “Wait on semaphore”
  - If not, figure out what they are
- High “Latch timeouts” could indicate low –spin

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

- **Keep “Flushes” at zero**
- **Progress uses “fuzzy checkpoint”**
  - **Too long to explain in detail…sorry**
  - **In a nutshell, all changes from the previous checkpoint must be written to disk before the end of the current checkpoint**
  - **Otherwise, *all transaction activity halts* until the DB catches up**

CKPT Length

- **Minimum 1 min under heavy load**
  - **Less than that and the APW's might not have time to do their job**
- **15 min or more during regular processing**

- **CKPT length is inversely proportional to BI cluster size**
  - **proutil <db> -C truncate bi –bi <size>**

Index Analysis

- proutil <db> -C idxanalys
- Look at « % Util » and « Level »
  - % Util = how full are the DB blocks
    - Keep this high (90%)
    - proutil <db> -C idxcompact 90
    - Watch out – bugs in idxcompact before 9.1E
  - Levels = depth of B-Tree
    - Idxbuild to reduce and rebalance tree
    - Watch on tables that have a lot of ADD/MOD/DEL
- Ignore tables with very small indexes

Table Analysis

- **proutil <db> -C tabanalys**
- **Use it to keep track of table growth**
  - **I have seen millions of records added "accidentally"**
- **"Scatter Factor" of limited value**
  - **IMHO…others may disagree**
- **Fragment count should equal record count**

_Tablestat and _indexstat

- **Gives usage statistics for tables and indexes**
- **Must set –basetable, -tablerangesize, -baseindex and –indexrangesize**
  - **By default DB only tracks first 50 tables**
    - **By file number, not alphabetically**

_Tablestat and _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
```


Server

- Three principal sub-systems
  - CPU
  - Memory
  - Disk
- All are easy to monitor
  - It's sometimes hard to find which process is abusing them!

CPU

- 4 type os CPU use
  - User, System, Wait I/O and Idle
- User: Important. Real work
- System: Operating system stuff
- Idle: Nothing to do…
- Wait I/O: Officially Wait I/O is idle time
  - Process *could* use the CPU if the I/O it was waiting for was available

CPU

- **Wait I/O not necessarily bad**
  - **Ex.: Alone on server doing heavy I/O.**
  - **CPU will show lots of I/O wait though really server is mostly idle**
- **Lots of WIO *could* indicate an I/O bandwidth problem**

- **Idle time is good**
  - **If more than 80% CPU during regular processing, your server will not handle peaks**

Memory

- **Don’t worry about “Free” memory**
  - **Systems like AIX never let mem sit idle**
- **DO worry about paging to page space**
  - **Absolutely ZERO paging allowed**
    - **Fix it immediately**
  - **Paging to file system is normal**

Memory

```
# 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
```

- **Pi/po on AIX**
- **Si/so on Linux**

Disks

- Next few slides are my opinion
- MANY will disagree
  - Few can argue effectively though
  - “this is how we’ve always done it” not a good reason
- YMMV – *Your Mileage May Vary*

Disks

- **Old days: BI on separate disk**
  - **Not so much anymore**
- **AI on separate disks**
  - **for recovery reasons, not performance**
- **Only consider separate AI/BI disks in special cases with very high write**

Disks

- **Today: Often (not always) combine all disks to form one big RAID 10**
  - **Segregating disks a waste of I/O BW**
  - **O.S. disks sometimes separate mirror**
- ***Assume* you need one big RAID 10 then *prove* why you don't**

The Truth about RAID 5

- **Hardware vendors sell you *disk space*, not I/O bandwidth**
  - ***SuperHyperNotRAID5_RAID5***
- **It's normal: they have their best interests in mind, not yours**

The Truth about RAID 5

- **RAID 5 is slower on writes only**
  - **More work to write (parity calculation)**
- **Manufacturers hide this behind disk cache**
  - **External array cache usually pretty big**
    - **IBM DS4xxx very fast**
  - **Internal RAID card cache usually small**
- **RAID 5 explained:**
  - **http://www.scottklarr.com/topic/23/how-raid-5-really-works/**

The Truth about RAID 5

- **In the real world:**
  - **FOUR times in the past four months, hardware upgrades led to decreased performance at my customers**
  - **All four times, many smaller disks replaced by few large disks in a RAID 5 configuration**
    - **I/O BW tanked**

RAID 5 Risks

- **Low-end implementations, cache not safe**
  - **Must disable on write**
  - **No more cache to absorb write peaks!**
- **Disk failure**
  - **Missing data calculated from parity**
    - **OUCH!!! That's slow**
  - **Entire array slow during rebuild**
  - **Any 2nd disk failure = array failure**

Disk Monitoring

- **Kb/sec not interesting**
  - **Look at I/O operations per second**
- **Expect 100-150 IOPS from modern disks (no cache)**
- **Watch out: With hardware RAID an O.S. disk is rarely a physical disk**

Disk Monitoring

```
# 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
```

Disk Monitoring

- If tps high for some disks, find out why
  - May not matter if disks virtual
    - Ex.: hdisk39, 40 and 41 striped across same 15 physical disks
- Avque & avwait = 0
- % busy sometimes false because of RAID

Nmon/Topas/Glance

- **Lots of O.S. monitoring tools**
- **I like nmon the best (AIX and RHEL)**
  - **Free**
- **HPUX people love glance**

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|UUUUUUssssssWWWWWWWWWWWWWWWWWWWWWW  >              |
 3     11.9   8.5  43.3  36.3|UUUUUssssWWWWWWWWWWWWWWWWWWWWWW   >               |
 4     21.9  10.9   8.5  58.7|UUUUUUUUUUssssssWWWW                    >         |
 5     28.4  12.4  25.9  33.3|UUUUUUUUUUUUUUssssssWWWWWWWWWWWW                 >|
 6     31.8   7.0   0.5  60.7|UUUUUUUUUUUUUUUUsss                      >         |
 7     13.9   3.0   0.0  83.1|UUUUUUs                                  >         |
                               +--------------------------------------------------+
       15.4   7.7  15.4  61.6|UUUUUUUsssWWWWWWW           >                     |
                               +--------------------------------------------------+
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
```

Wrap-Up

- **Remember: Identify a specific problem and apply *one* specific solution**
- **Document everything**

- **Consider using a monitoring tool with trend history**
  - **Can go back in time and compare metrics**

Questions?

Progresswiz

Progresswiz Consulting

- **Questions or comments?**
  pk@progresswiz.com
- **PPTs, articles and tools available at**
  www.progresswiz.com

Merci !

Progresswiz