PUGCHALLENGE AMERICAS EXCHANGE

Tales of the Secret Bunker
2017 Edition

Gus Björklund,

Head Groundskeeper, The Parmington Foundation,

Mike Furgal

MFWIC of MDBA, Progress Professional Services

Americas PUG Challenge
Manchester, NH, USA
4-7 June 2017

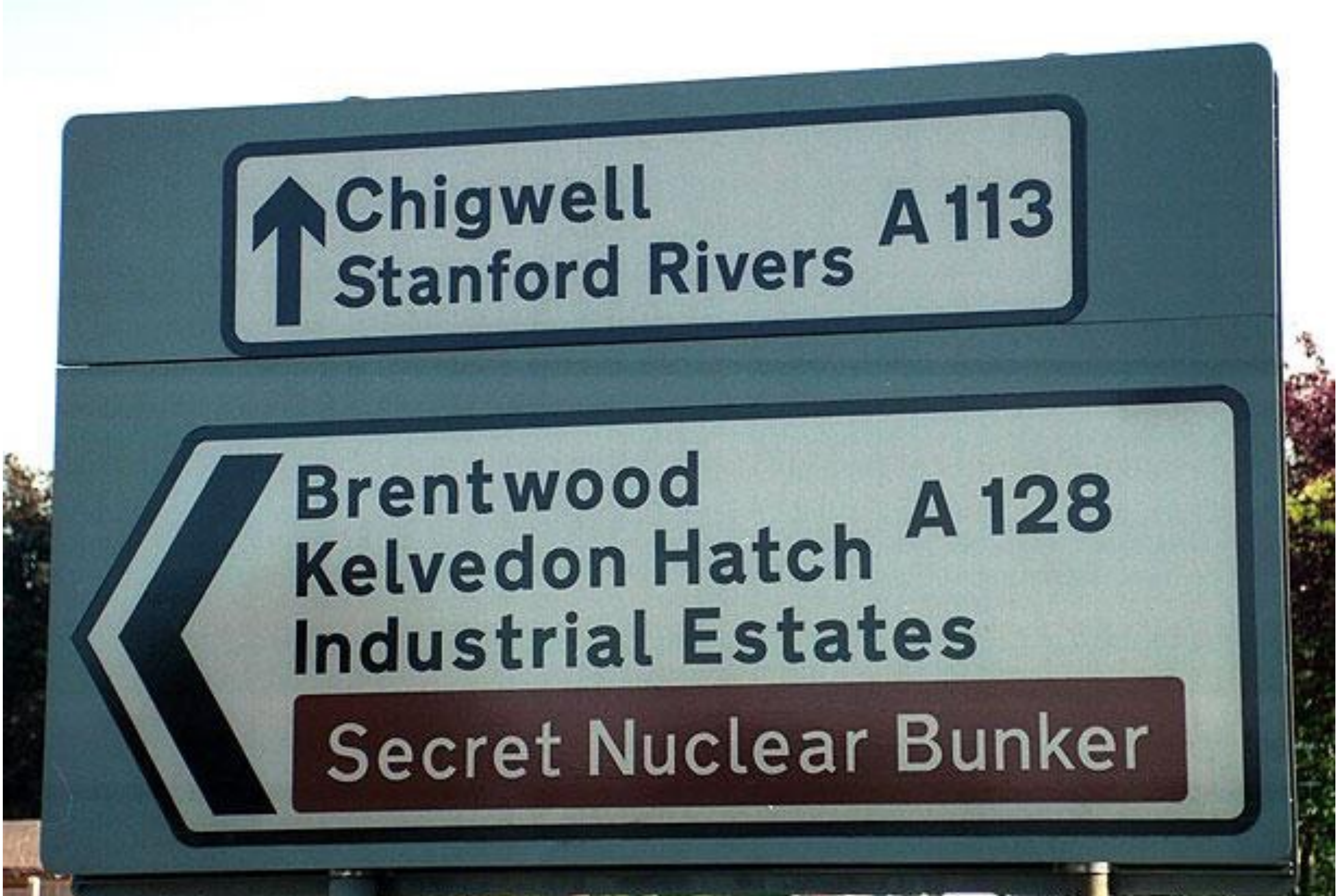
Abstract

In this talk we share the results of numerous database performance measurements taken with modern releases of the OpenEdge RDBMS. These measurements were conducted in the well-known Secret Bunker.

In particular, we will show the effects of combining certain database features. Among these are After-Image Journalling, Transparent Data Encryption, OpenEdge Replication and a few other things.

Join us as we go down into the Secret Bunker to see what we can find there.

how to get to the bunker



Notices

- Please ask questions as we go
- We have not tested everything.
- YMMV (Your mileage may vary, transportation, meals, and accomodations not included).

Experience, next, to thee I owe,
Best guide; not following thee, I had remain'd
In ignorance; thou open'st wisdom's way,
And giv'st access, though secret she retire.

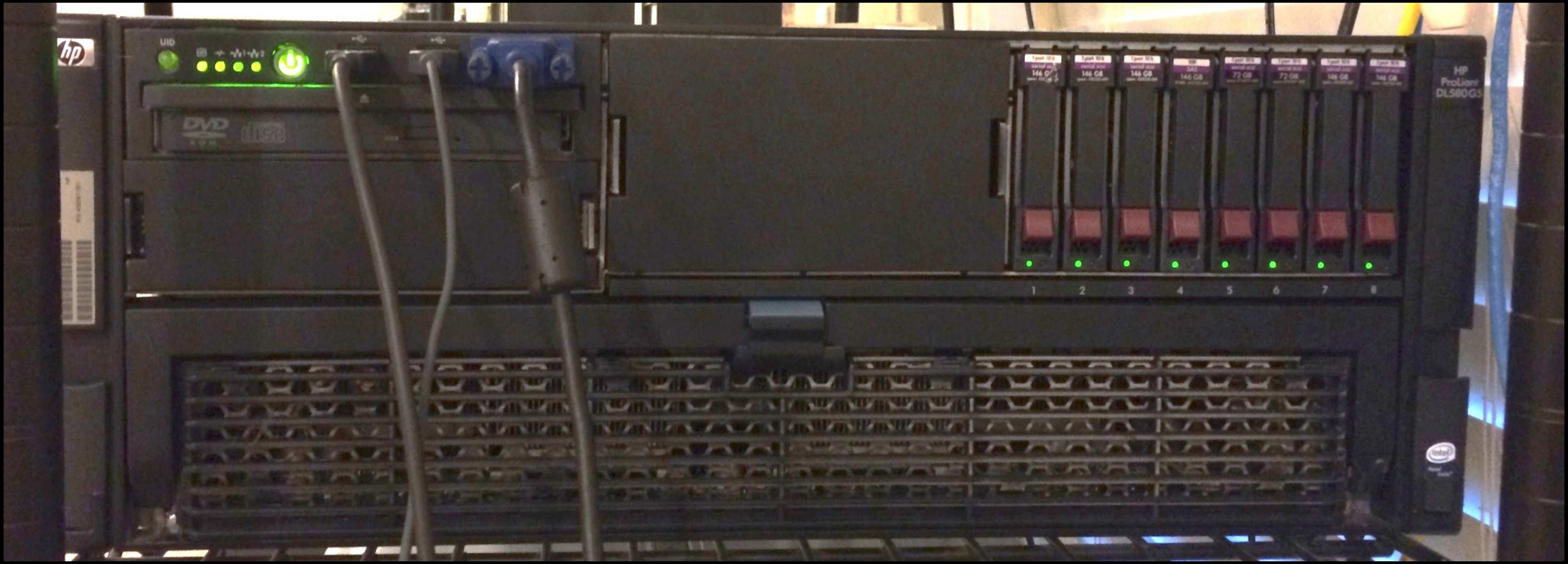
John Milton, *Paradise Lost*

We have lots of experience to share.

test machine number 1

aka bunker15

bunker15


UID
DVD
HP
ProLiant
DL580G5
146 GB
72 GB
1
2
3
4
5
6
7
8

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

the bunker15 machine - an HP box

- 4 quad-core 2.4 GHz Intel processors
  - 4800.25 bogomips
- 64 GB memory
- 16 x 300 GB 10,000 rpm drives
  - RAID 10 for /home
- Centos 6 Linux (2.6.32-504.12.2.el6.x86_64)
- OpenEdge 11.5.1 and 11.7
- ATM 7

New this machine cost $35,000 USD.

Used we found it for $3,500 USD

The "good old ATM benchmark"

ATM refresher

- Standard Secret Bunker Benchmark
  - baseline config always the same since Bunker #2
  - Not today, though – 9x larger database: 100 GB
- Simulates ATM withdrawal transaction
- 150 concurrent users
  - execute as many transactions as possible in given time
- Highly update intensive
  - fetch 3 rows
  - update 3 rows
  - create 1 row with 1 index entry

let's run some tests

build a 7200 tps database

using 4 concurrent "loaders"

7200 tps ATM database has

account rows	720,000,000
teller rows	720,000
branch rows	72,000
total size	100 GB

build time was 8 hours 59 minutes

720,792,000 records created

(1,337,276 records per minute)

Test 1: run the standard ATM baseline

About ATM ... the standard baseline config

```
-maxAreas 90      # maximum storage areas
-n 250            # maximum number of connections
-L 10240          # lock able entries
-B 64000          # main buffer pool number of buffers
-spin 5000        # spinlock retries
-bibufs 32        # before image log buffers
```

```
                    300       600       900      1200      1500
Clients   Tps |--------|---------|---------|---------|---------|
   150 1087.6 ************************************
   150 1307.4 ********************************************
   150 1315.6 ********************************************
   150 1332.8 ********************************************
   150 1316.0 ********************************************
   150 1313.1 ********************************************
   150 1320.7 ********************************************
   150 1314.3 ********************************************
   150 1287.4 *******************************************
Clients   Tps |--------|---------|---------|---------|---------|
standard baseline average is 1287 tps
```

1287

Is that a good result?

Why?

Test 1.1: run standard ATM baseline

add lruskips 100,
add napmax 100,
change spin from 5,000 to 50,000

```
                    500       1000      1500      2000      2500
Clients    Tps |--------|---------|---------|---------|---------|
   150 2368.6 ***********************************************
   150 2430.1 *************************************************
   150 2406.1 ************************************************
   150 2400.6 ************************************************
   150 2444.7 *************************************************
   150 2407.0 ************************************************
Clients    Tps |--------|---------|---------|---------|---------|
std BL    1287 ************************

average is 2409 tps, double the standard baseline
NOTE SCALE CHANGE !!!

THIS IS OUR NEW BASELINE !!!
```

1287 to 2409

Twice more better with 3 easy changes !

Test 2: modified ATM baseline

add after-image journalling

(9 variable extents on other disks)

archive daemon runs every 120 sec

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 1623.5 ****************
   150 1666.9 *****************
   150 1646.8 ****************
Clients   Tps |--------|---------|---------|---------|---------|
new BL 2409   ************************

average is 1645 tps, 68 %
```

Hey !!!!

Not so fast, experts. Think we are stupid?

Didn't you tell us before that

"After-image journalling overhead is negligible." ?

2409 to 1645 is NOT negligible.

Test 2.1: modified ATM baseline

with after-image journalling

add after-image journal writer (AIW)

```
Clients   Tps |--------|---------|---------|---------|---------|
   150 2417.0 ************************
   150 2335.2 ***********************
   150 2417.5 ************************
   150 2383.2 ************************
   150 2335.4 ***********************
   150 2273.4 ***********************
Clients   Tps |--------|---------|---------|---------|---------|
new BL 2409   ************************

average 2360 tps, 98 %
```

We will take 2360 tps
as the new, modified baseline.

for the remaining tests

Test 3: modified ATM baseline

with after-image journalling

enable encryption (no policies)

```
#! /bin/sh
#
echo 'e "Encryption Policy Area":30,64;8  ./db' \
>tdepolicyarea.st
#
prostrct add atm tdepolicyarea.st
proutil atm -C enableencryption \
       -Autostart admin << EOF
AaBbCc1!
AaBbCc1!

EOF
#
```

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 2305.9 ***********************
   150 2299.2 ***********************
   150 2319.9 ***********************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************

average is 2307, 98 %
```

Test 3.1: modified ATM baseline

with after-imaging

with encryption enabled

add policies

```
#! /bin/sh
#
for T in account branch teller history1 history2
history3 history4
do
    proutil atm -C epolicy manage table encrypt "$T"
done
#
for I in account.account branch.branch\
         teller.teller \
         history1.histid history2.histid \
         history3.histid history4.histid
do
    proutil atm -C epolicy manage index encrypt "$I"
done
```

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 2288.3 ***********************
   150 2328.8 ***********************
   150 2291.7 ***********************
   150 2304.5 ***********************
   150 2293.9 ***********************
   150 2288.1 ***********************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 2298 tps. 97 %
```

Test 3.2: modified ATM baseline

with after-imaging

with encryption

with policies

add 14 proutil encryptors

(6 for tables, 6 for indexes)

```
#! /bin/sh
#
for T in account branch teller history1 history2 \
         history3 history4
do
    proutil atm -C epolicy manage table update "$T" \
             2>&1 >>encrypt_tb.log &
done
#
for I in account.account branch.branch teller.teller \
         history1.histid history2.histid \
         history3.histid history4.histid
do
    proutil atm -C epolicy manage index update "$I" \
             2>&1 >>encrypt_ix.log &
done
```

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 1003.8 **********
   150 1033.5 **********
   150 1039.1 **********
   150 1078.7 ***********
   150 1085.2 ***********
   150 1113.0 ***********
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 1058 tps, 45 %
```

2298 to 1058

WTF ?

Test 3.3: modified ATM baseline

with after-imaging

with encryption

with policies

fully encrypted

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 2472.8 *************************
   150 2402.6 ************************
   150 2508.5 *************************
   150 2400.7 ************************
   150 2397.3 ************************
   150 2392.4 ************************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 2428 tps, 103 %
```

1058 to 2428

2438 > 2360

Why was it faster
than the reference ?

Test 3.4: modified ATM baseline

with after-imaging

with encryption

with policies

fully encrypted

stop db, flush filesystem cache, start db

```
proshut database/atm -by

sync

echo 3 > /proc/sys/vm/drop_caches

proserve -pf atm.pf
```

after flushing filesystem cache

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 1823.4 ******************
   150 2240.7 **********************
   150 2540.8 *************************
   150 2578.7 **************************
   150 2368.3 ************************
   150 2499.2 *************************
   150 2362.8 ************************
   150 2363.6 ************************
   150 2316.9 ***********************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************
average of last 4 is 2381 tps, 101 %
```

Test 4: modified ATM baseline

with after-imaging

with encryption

with policies

fully encrypted

upgrade from 11.5.1 to 11.7

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 2299.4 ***********************
   150 2282.2 ***********************
   150 2299.1 ***********************
   150 2270.0 ***********************
   150 2275.1 ***********************
   150 2253.8 ***********************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 2281 tps, 97%
```

Test 4: modified ATM baseline

with after-imaging

with encryption

with policies

fully encrypted

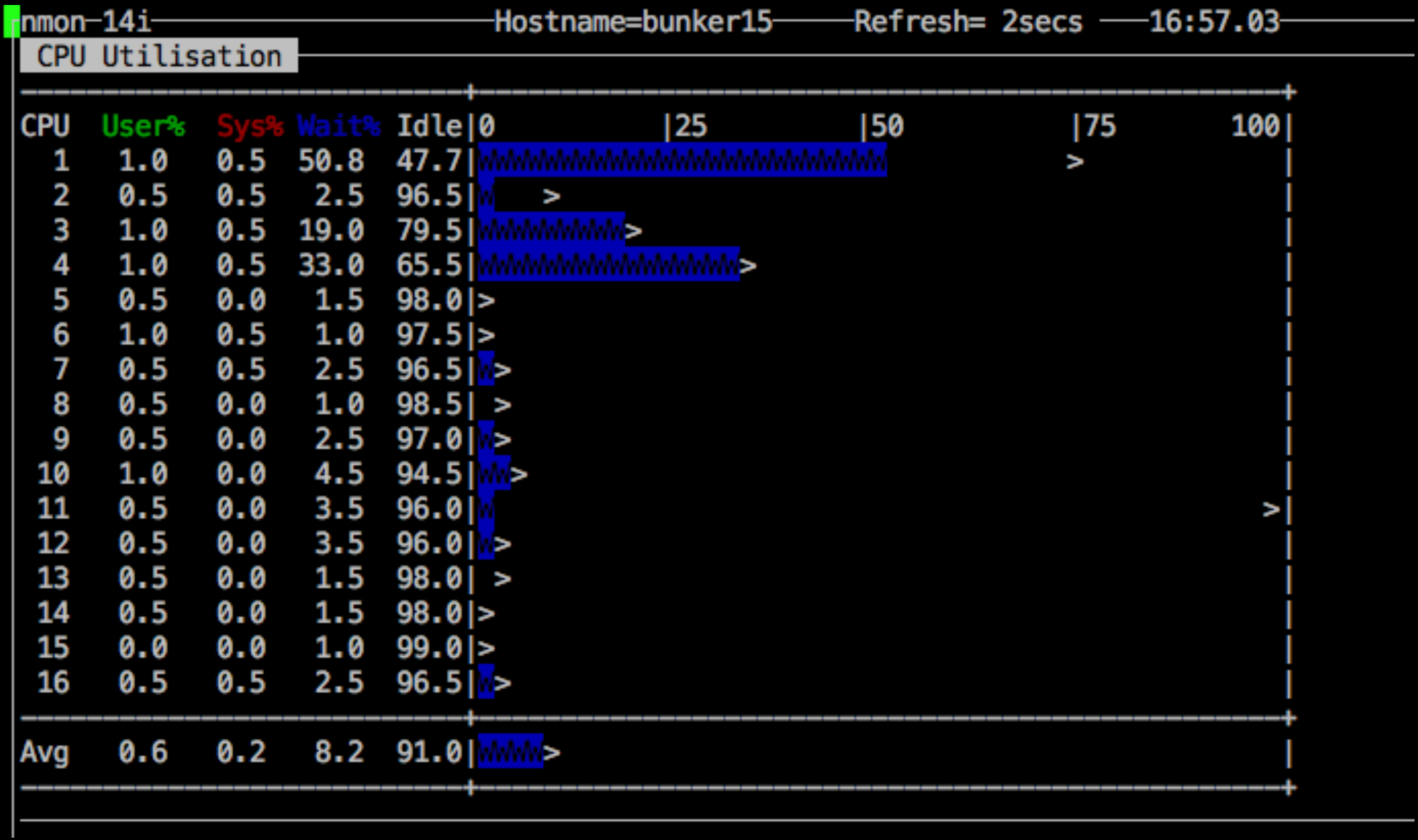
add OpenEdge Replication

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150  170.6 **
   150  166.2 **
   150  165.9 **
   150  165.5 **
   150  167.5 **
   150  167.8 **
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************

average is 167 tps, 7 % !!!!!
```

a complete disaster !

photo from popular mechanics



```
nmon-14i─────────────────────Hostname=bunker15─────Refresh= 2secs ───16:57.03─────────
 CPU Utilisation ─────────────────────────────────────────────────────────────────────
---------------------------+--------------------------------------------------+
CPU  User%  Sys% Wait% Idle|0           |25         |50          |75       100|
  1   1.0   0.5  50.8  47.7|WWWWWWWWWWWWWWWWWWWWWWWWW           >             |
  2   0.5   0.5   2.5  96.5|W   >                                             |
  3   1.0   0.5  19.0  79.5|WWWWWWWWW>                                        |
  4   1.0   0.5  33.0  65.5|WWWWWWWWWWWWWWWW>                                 |
  5   0.5   0.0   1.5  98.0|>                                                 |
  6   1.0   0.5   1.0  97.5|>                                                 |
  7   0.5   0.5   2.5  96.5|W>                                                |
  8   0.5   0.0   1.0  98.5| >                                                |
  9   0.5   0.0   2.5  97.0|W>                                                |
 10   1.0   0.0   4.5  94.5|WW>                                               |
 11   0.5   0.0   3.5  96.0|W                                                >|
 12   0.5   0.0   3.5  96.0|W>                                                |
 13   0.5   0.0   1.5  98.0| >                                                |
 14   0.5   0.0   1.5  98.0|>                                                 |
 15   0.0   0.0   1.0  99.0|>                                                 |
 16   0.5   0.5   2.5  96.5|W>                                                |
---------------------------+--------------------------------------------------+
Avg   0.6   0.2   8.2  91.0|WWWW>                                             |
---------------------------+--------------------------------------------------+
──────────────────────────────────────────────────────────────────────────────────────
```

we are completely disk bound

source and target are on the same machine

this was a bad idea.

Test 4: modified ATM baseline

with after-imaging

with encryption

with policies

fully encrypted

move target to Norcross, GA
(1 MB/sec line)

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150  722.2 *******
   150  676.7 *******
   150  668.4 *******
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************

average is 688 tps, 29%
```

Can we do better ?

How ?

Yes !

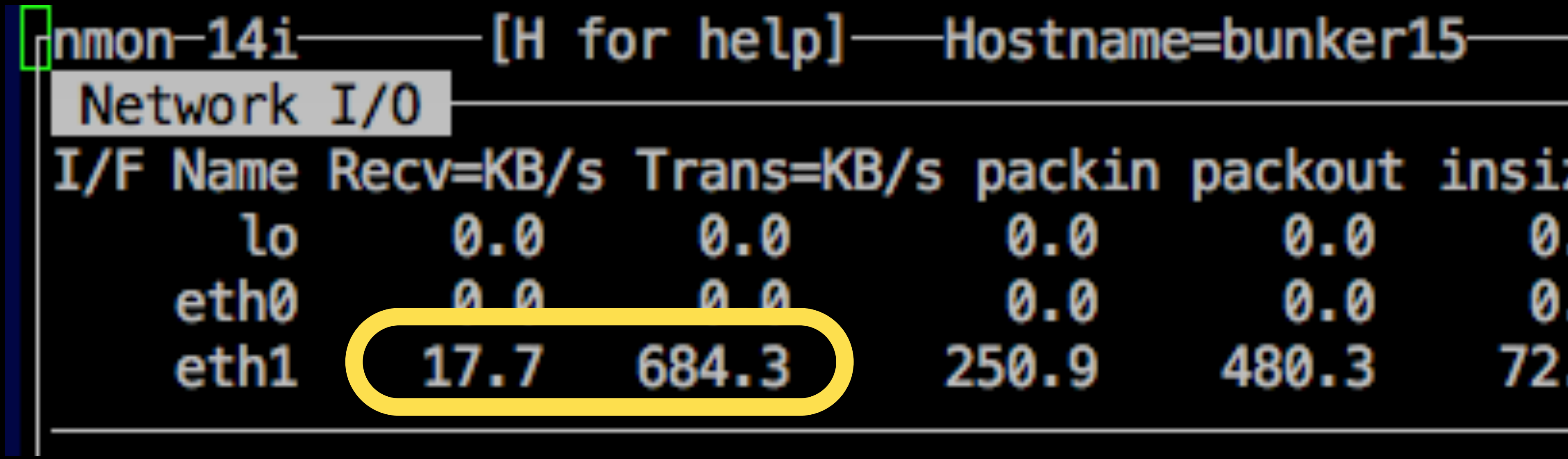
Raise -pica to 1,000,000

(about 977 MB of queue space,

default is 64 KB)

```
                   1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150 2226.0 **********************
   150 2015.2 ********************
   150 2197.4 **********************
   150 2127.2 *********************
   150 2174.0 **********************
   150 2080.4 *********************
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 2136 tps or 91%
```



```
┌nmon─14i──────────[H for help]───Hostname=bunker15──────
│ Network I/O ───────────────────────────────────────────
│I/F Name Recv=KB/s Trans=KB/s packin packout insi
│      lo      0.0       0.0       0.0      0.0     0.
│    eth0      0.0       0.0       0.0      0.0     0.
│    eth1     17.7     684.3     250.9    480.3    72.
│────────────────────────────────────────────────────────
```

What does our experience tell you?

Results summary part 1

nr.	description	tps
1	simple baseline	1287
1.1	simple baseline + lruskips 100, napmax 100 spin 50,000	2471
1.2	simple baseline + lruskips 100, napmax 50, spin 50,000, bibufs 64	2405
2	add after-image journalling	1645
2.1	add after-image writer (AIW)	2360

Modified BL with AI is 2360

Results summary part 2

nr.	description	tps
3	enable encryption, no policies	2307
3.1	add policies (default cipher AES-128-CBC)	2298
3.2	add 12 proutils to encrypt tables and indexes	1058
3.3	after everything is encrypted	2428
3.4	flush caches, do 3.3 again	1823
3.5	3.3 again after caches filled	2389
4.0	3.3 with OpenEdge Replication	167
4.1	3.3 with OE Replication, 2'nd try	2136

Now, lets try a small machine

test machine nr. 2

aka bunker17

bunker17 - a Dell PowerEdge 2950 circa 2009

- 4 dual-core 3.0 GHz Intel processors
- 32 GB memory
- PERC 6i RAID controller
- 6 Hitachi 7200 rpm SATA drives
  - 4 RAID 0
  - 2 JBOD for bi and ai
- Centos 7 Linux
- OpenEdge 11.5.1
- ATM 7

Total cost $555.26 USD

bunker17

Name	Qty	Cost
Computer	1	$189.00
Disk drives	6	$300.00 (@ $50 ea)
Drive mounting screws	24	$6.32 for 50
SAS Drive carriers	6	$29.94 (@ $4.99 ea)
SAS to SATA adapters	6	$30.00 (@ $ 5.00 ea)
Operating system	1	$0.00
Total	44	$555.26

bunker17 machine

Standard Baseline +

-lruskips 100

-napmax 100

```
                  1000      2000      3000      4000      5000
Clients   Tps |--------|---------|---------|---------|---------|
   150  409.8 ****
   150  422.6 ****
   150  425.4 ****
   150  431.7 ****
   150  452.1 *****
   150  459.2 *****
Clients   Tps |--------|---------|---------|---------|---------|
BL w ai  2360 ***********************


average is 433 tps
```

What would a brand new machine like bunker 15 cost?

bunker17

Name	Qty	Cost
Computer	1	
Disk drives	16	
Power supply, DVD, etc.	1	included
Operating system	1	$0.00
3 years service	1	included
Total	18	$11,439.28 USD

Lessons**

0) lruskips has a big impact

1) napmax has a lesser impact

2) after-image overhead is essentially zero

3) encryption overhead is essentially zero

4) 11.5.1 and 11.7 database performance are the same

5) OpenEdge Replication overhead is about 9 %

6) good hardware is very inexpensive

*** YMMV (your mileage may vary).*
transportation, meals and accomodations not included.

"Experience is a brutal teacher because
she gives the test first and the lesson afterwards."

Vernon Sanders Law

Want Answers?

email:

- gus642@gmail.com
- mfurgal@progress.com