Test Results
PostgreSQL V7.4.7 versus V8.0.3
For the
RFC Archive System[1]

August 8, 2005

1.0 Summary

Test results indicate that both versions of PostgreSQL are faster than the Informix relational database management system (RDBMS) as it is currently configured on the RFC Archive Systems (RAX) at the 13 River Forecast Centers (RFC). This result is not a surprise, what is surprising is how much faster PostgreSQL is compared to Informix. This improved performance addresses one of the primary concerns the RFCs have with the RFC Archive Database.

The other concern expressed with the move to PostgreSQL is the loss of Informix's table fragmentation feature, a feature that improved query performance on large tables. The loss of this feature is no longer a concern as a result of this testing.

Based on this testing, PostgreSQL V8.0.3 is approximately 1.5 times faster than V7.4.7. Because performance (i.e. "speed") is a major concern of the RFCs, it is recommended that PostgreSQL V8.0.3 be used for the RAX ob7 RDBMS.

It should be noted that these results are for the RFC Archive System only, which is a dedicated system. The results will most likely not translate to the performance of the IHFS database, that resides on a system that has multiple tasks and other databases in the PostgreSQL RDBMS.

It is also recommended that additional testing of the PostgreSQL RFC Archive databases be done once applications have been rewritten and are available to test with. This additional testing will give a better feel for performance when the system is running in a set-up more similar to the field and will also allow for tuning of the RDBMS.

[1] Report written by A. Juliann Meyer, RAXUM Team Leader, Sr. Hydrologist – Data Systems, Missouri Basin River Forecast Center, Pleasant Hill, MO

2.0 Introduction

2.1 Background

Plans call for the AWIPS system to upgrade the operating system (OS) from RH7.2 to RHE3.0 in build ob6. When OHD/HL was setting up a RAX with RHE3.0 in February 2005, it was discovered that the IBM Informix IDS 9.3.UC1 would not work with the new operating system. The long-range plans were to move the RFC Archive Database to PostgreSQL (the operational AWIPS databases will be moved in build ob6) in a later build in order to benefit from lessons learned moving the operational databases. Due to the OS conflict, the migration was moved up and the RAXUM team and OHD/HL determined that for one build, ob6, the RAX would remain with RH7.2 and the Informix RDBMS. This gives the RAXUM team and OHD/HL less than a year to complete this significant and complex task (the conversion must be completed by late March 2006). A timeline of sub-tasks to meet this deadline can be found on the RAXUM team's website (http://www.nws.noaa.gov/oh/rfcdev/projects/rfcADEMT_chart.htm).

In the initial planning phase, it was discussed whether PostgreSQL version 7.4.7 or 8.0.3 should be used on the RAX. The IHFS DB and other operational databases that migrate to PostgreSQL in build ob6 will be using version 7.4.7, as version 8.0.3 was not released in time for this build. However, version 8.0.3 is available for the RAX migration, and the PostgreSQL consultant indicated that version 8.0.3 would provide better performance than version 7.4.7.

"Speed" is, and always has been, one of the primary concerns because of the large quantities of data that a RFC could be storing in the RFC Archive DB. The RFCs, in general, have never been happy with the performance of Informix RDBMS, and the performance of Informix on the RAX has not met some RFCs expectations.

Another concern has been the loss of the Informix table fragmentation feature that had helped somewhat with the "speed" issue on the RAX. Selected tables were fragmented in the RFC Archive DB as a result of a past study at the Missouri Basin River Forecast Center (MBRFC). That study had shown that the use of table fragmentation could significantly speed up query performance.

Thus, it was determined that a test of the two PostgreSQL versions should be conducted.

2.2 Purpose

The purpose of this test is to provide information on the performance of PostgreSQL versions 7.4.7 and 8.0.3. The test was also run on the RAX at MBRFC for a "baseline/current set-up" comparison.

3.0 Description of Systems

3.1 Hardware

The testing was performed on two systems, ax-krf for the "baseline/current set-up" Informix run and ax2-nhdr for both versions of PostgreSQL runs. Both systems have the following hardware:

- Dedicated system, Rack mounted
- Intel Xeon 2.4GHz/400MHz
- 2 - 512MB PC2100 CL2.5 ECC DDR SDRAM RDIMM
- Ultra 320, ServeRAID-5i SCSI Controller (single channel)
- Six 73.4GB 10K rpm Ultra160 SCSI HS
- 10/100/1000 Port Ethernet Server Adapter
- Tape drive - 40/80GB DLTVS HH Int. SCSI Drive (Half-High) and Ultra 160 PCI Adapter (required for Tape device when using ServeRAID5i
- DVD Drive/Recorder - DVR-A04 Pioneer DVR (4.7gb)

3.2 Software

The differences between ax2-nhdr and ax-krf are the RDBMS and the operating system.

ax2-nhdr

RDBMS: PostgreSQL versions 7.4.7 and 8.0.3; both versions have been installed but only one version can be running at any given time. Uses cooked file space for the database. No logging and no replication.

OS: Red Hat Enterprise 3.0

ax-krf

RDBMS: IBM Informix IDS 9.3 UC1. Uses cooked file space for the database. No logging and no replication.

OS: Red Hat 7.2

4.0 RFC Archive DB Move from Informix to PostgreSQL

4.1 The Changes

Changes were made to the RFC Archive Database schema based on what could and could not be carried over to PostgreSQL from the current RFC Archive DB as it resides in Informix. The following changes were made:

- The following tables (which took advantage of Informix's table fragmentation) are no longer fragmented: *pecrsep*, *pedrsep*, *peoosep*, *pedfsep* and *unkstnvalue*.
- The parameters of "extent size, next size" no longer exist when defining a table.
- The methology on lock mode changes with the move to PostgreSQL.
- For the following tables the column name "desc" was changed to "descr": *opersnow17*, *segoper*, *opersacsma*, *operunithg*, *operapicont*, *area*, *areasens*, *cgroup*, *drain*, *fgroup*, *opertype*, *pos* and *seg*.
- For table *crest*, the column "old" was changed to "olddatum".
- Table *riverstat* is removed in ob7; this table has been obsolete since ob5.
- Table *rating* changed in ob7 to accommodate a request from the National RATINGS team led by Dave Reed. Part of the change takes advantage of a feature that exists in PostgreSQL but does not in Informix, i.e. having a column as an array.
- Table *ratingshift* has a couple of changes due to the new rating table structure, these are: added column "src" and changed the primary key to be in sync with then *rating* table.
- Table *rivercrit*, selected columns changed from "float" to "numeric(10,2)". These columns are: fis, action, alert, bank, flood, modflood, majflood and record.
- For table *pempsep* the column name "obstime" was changed to "cal_yr"... this makes it sync with the *pemrsep* table name for that column.
- For the SHEF data value tables, the column name "value" was changed to "datavalue".
- For the NWSRFS mod tables, the column name "value" was changed to "modvalue".
- For the table *prod*, the column name "max" was changed to "pmax".
- Most of the changes related to data type were straight forward, however the data type used in Informix for date-time columns in several tables was not available in PostgreSQL. A different date-time data type was chosen for these columns, therefore some of the data unloaded from Informix had to be reformatted before it could be loaded in PostgreSQL. The changes are as follows:
  - Table *pedpsep*, the column "obstime" (which was "datetime year to month") becomes data type of DATE, where the day is always 01.

- Table *ingestfilter*, the column “obstime” (which was “datetime hour to second”) becomes data type of TIME.
- For table *qadjust* the columns “b_date” and “e_data” (which were data type “datetime year to month”) become data type of DATE, where the day is always 01.
- For tables *statesapicont*, *statessacsma* and *statessnow17*, the column “obstime” (which was data type “datetime year to hour”) becomes data type of TIMESTAMP, where minutes and seconds are always 00:00.
- For the NWSRFS mod tables, when applicable, the columns “sdate”, “rdate”, “vdate” and “edate” (which were data type “datetime year to hour”) become data type of TIMESTAMP, where minutes and seconds are always 00:00.
- For tables *datalimits* and *locdatalimits*, the columns “monthdaystart” and “monthdatend” (which were data type “datetime month to day”) become varchar(5).

4.2 Data for the Test Databases

Two test databases were defined, and selected data from MBRFC’s RFC Archive DB was loaded into the test databases. Some of the SHEF Data Value tables in MBRFC’s Archive DB have more than 5 million rows in them. For a handful of stations data goes back to 1983, while the majority of the data is from Fall 1994 to present. Selected data for late spring and summer of 1993 is also stored.

Because of a limitation of Informix on file size when writing to disk, the dbexport command could not be used to unload the tables in the RFC Archive Database on ax-krf. Instead a method provided by CBRFC was used to unload the tables one by one. The procedure used was:

1. Make a named pipe (Note: only have to do this step one time)
   - mkfifo informix_pipe
2. Start the gzip process
   - gzip < informix_pipe > tablename.unl.gz &
3. Use dbaccess or a script and instead of unloading to a file, unload to the pipe. Note that no “order by” clause is included.
   - unload to informix_pipe select * from “*table name*”, where “*table name”* is one of the RFC Archive DB tables

The following tables were unloaded from the Informix RFC Archive database: location, ingestfilter, rivercrit, aliasid, huc2, huc4, huc6, huc8, wfo_hsa, rfc, shefpe1, shefpetrans, shefdur, shefts, shefex, shefpe, shefprob, shefqc, country, state, counties, modctrl, datalimits, locdatalimits, pairedvalues, pedcsep, pedpsep, peqfsep, pehpsep, pemrsep, pedrsep, pedfsep, pecrsep, peoosep, unkstnvalue, vfyruninfo and vfypairs.

These unload files were moved to the nhdr system. A sed script provided by OHD/HL was used to strip off the ending pipe symbol, "|", in the Informix unload files, and an awk script was used to modify the unload file for the pedpsep table because of the change to the datetime field. The psql copy command was then used to load the data files into the appropriate tables of the two PostgreSQL test databases.

5.0 Testing Procedure

Support programmers for selected RAX baseline applications provided sample queries from that the applications they support. The queries came from the following applications: the level 1 processors, datview and the national river verification software. Since some of the queries used Informix specific sql commands that are not available in PostgreSQL, the queries were modified as needed to create the query set for the PostgreSQL testing. Changes that had to be made are as follows:

- ✔ Any query that used the command *matches* was modified to use *like*.
- ✔ Any query that used the command *unique* was modified to use *distinct*.
- ✔ All queries have a semi-colon added at the end of the query.
- ✔ Any double-quotes used in the queries were replaced with single-quotes.
- ✔ Changes to "where clauses" were made based on any changes to tables as described in section 4.0. These changes include column name changes and data type changes.
- ✔ The following portion of query 23 was changed from

```
(validtime - basistime) > 0 UNITS HOUR AND
(validtime - basistime) <= 72 UNITS HOUR
```

to

```
EXTRACT(HOUR FROM validtime – basistime) > 0 AND
EXTRACT(HOUR FROM validtime – basistime) <= 72
```

In all, the test script cycles thru 31 queries 100 times each. Appendix A contains copies of the test scripts. A listing of the actual queries run by these test scripts can be found in Appendix B. An effort was made to take cache out of play by requesting data for different stations, data types and time periods and changing the order of the queries in the test scripts. In fact, after the Informix test run, the order of the queries was adjusted for the PostgreSQL test runs; queries #29 and #30 were moved to a different part of the test script. This adjustment was made due to suspicions that cache may have improved query #31 results.

For the Informix test, no database cron jobs were running and the RAX shefdecoders (raw and processed) were off when the test script was run on the ax-krf system. However some non-database cron jobs might have executed while the test script was running. The script was also run when the raw shefdecoder was running in the late

afternoon and evening, and the resulting run times were very similar to when the shefdecoders were off.

For the PostgreSQL testing, since none of the RAX applications have been converted to access PostgreSQL, no applications were running at the time the test scripts were run on ax2-nhdr. No cron jobs were running on ax2-nhdr at the time of the testing.

At the time the testing was done on ax-krf and ax2-nhdr, the only user accessing the systems was the person doing the testing.

Both test scripts used the system clock and the "date +%s" command to measure the run time. The date command was run immediately before and after the execution of each query. We know that this method is not perfect, has limitations and is a relative measure of the run times. However, it does allow the same method of run time calculation for both Informix and PostgreSQL.

An excerpt of the logfile that is created by a test script follows.

```
run,query,start,end
run 1,q1,1121787934,1121787934
run 1,q2,1121787934,1121787934
run 1,q3,1121787934,1121787935
run 1,q4,1121787935,1121787935
run 1,q5,1121787935,1121787935
run 1,q6,1121787935,1121787935
run 1,q7,1121787935,1121787935
run 1,q8,1121787935,1121787935
run 1,q9,1121787935,1121787935
run 1,q10,1121787935,1121787935
run 1,q30,1121787935,1121787935
run 1,q29,1121787935,1121787945
run 1,q11,1121787945,1121787945
```

6.0 Test Results

The three logfiles from the test runs were imported in to MS Excel for analysis. Results for various factors are shown in Tables 1, 2 and 3. More detailed summary for both the Informix and PostgreSQL testing can be found in Appendix C. Note that in queries #28, #29 and #30, #31 (queries with and without an "order by" clause) the impact of adding an "order by" clause is hardly noticeable in PostgreSQL.

Table 1.

Total Script Run Time		
Informix V9.3 UC1	**PostgreSQL VT7.4.7**	**PostgreSQL V8.0.3**
32522 sec. (9.03 hrs)	2572 sec. (42.86 min)	1654 sec (27.57 min)

Table 2.

Average Run Time for 1 Cycle		
Informix V9.3 UC1	**PostgreSQL VT7.4.7**	**PostgreSQL V8.0.3**
325.15 sec	25.64 sec.	16.49 sec

Taking a closer look at the run times for individual queries, it can be seen that in the Informix test two queries, #28 and #29, take up 73% of the average total run time for one cycle. In fact, these two queries take up 80% of the PostgreSQL V7.4.7 average run time and 66% of PostgreSQL V8.0.3 run time. Table 4 shows the combined average run times for these two queries and Table 5 shows the average run time for one cycle excluding these two queries for each of the tests. Note that the difference in average run times for one cycle for the two versions of PostgreSQL decreases significantly; the difference drops to less than 1 second. It is PostgreSQL V8.0.3 performance on queries #28 and #29 that really allows it to shine, V8.0.3 only takes half the time that it takes V7.4.7 to perform the same two queries. These queries, #28 and #29, both ask for a large quantity of data from a table that has over 11 million rows in it. We believe that this type of query will be commonplace as more data is stored in the RAX archive DB.

This demonstrates that the type of query and how a RDBMS handles the queries can impact performance. How the RDBMS is configured also can have an impact. In the case of Informix on the RFC Archive System, the suggested settings as delivered in the onconfig.std file were used … no additional tuning has been done. Appendix D contains a copy of the onconfig file uses on ax1-krf. The default values for configuration were used for both versions of PostgreSQL on ax2-nhdr. Appendices E and F have listings of the configuration parameters for both versions.

It is possible that some non-DB cron jobs by users oper and root might have run during the 9+ hours that it took to run the Informix test on the ax-krf system. Since the ax-krf system is operational at MBRFC, it was not possible to eliminate all non-DB activity, but the possible impact of such activity was minimized. In contrast, the PostgreSQL testing

on ax2-nhdr was done under ideal conditions, no one else was on the system and no cron jobs were running.

Another factor that may have played some role is how the databases were populated. All of the data in the RFC Archive Database on ax-krf in the SHEF Data Value tables was posted to the database by the raw or processed shefdecoders, and all the data in the vfypairs table was posted by the verify software. Thus the operational database has grown over time, and chunks have been added to the dbspaces as needed to accommodate this growth. With Informix, the number extents that a table has can affect performance, a review of the extents on the ax-krf system indicates that "too many extents" (i.e. greater than 16) may have impacted the performance of some of the test queries to tables *pecrsep*, *pedcsep*, *vfypairs*, *pedrsep* and *pedfsep*. In contrast, data in both PostgreSQL test databases was loaded at one time from pre-existing files, so there has been no growth over time in these tables. In general, with PostgreSQL extents are not an issue.

Since the RFC Archive System is dedicated to archiving (the RDBMS, its software and the flat file archive), the performance should be better with PostgreSQL than the current RAX set-up with Informix, even if the performance isn't as spectacular as these test results indicate.

Table 3 is on the next page.

Table 4.

Sum of Average Run Time for queries 28 and 29		
Informix V9.3 UC1	**PostgreSQL VT7.4.7**	**PostgreSQL V8.0.3**
237.07 sec	20.47 sec.	10.81 sec

Table 5.

Average Run Time for 1 Cycle excluding #28 and #29		
Informix V9.3 UC1	**PostgreSQL VT7.4.7**	**PostgreSQL V8.0.3**
88.08 sec	5.17 sec.	5.68 sec

Table 3.

	Average Run Time (seconds) for each Query		
Query	Informix V9.3 UC1	PostgreSQL V7.4.7	PostgreSQL V8.0.3
1	0.2	0.02	0.03
2	0.26	0.02	0.01
3	1.46	0.31	0.64
4	4.71	0.06	0.11
5	0.18	0.02	0.04
6	0.07	<0.01	0.03
7	0.09	0.01	0.02
8	0.16	0.04	<0.01
9	0.08	0.01	0.01
10	0.09	0.02	0.02
11	0.03	0.03	0.03
12	3.8	0.07	0.05
13	3.61	0.06	0.03
14	0.14	0.03	0.01
15	1.99	0.09	0.08
16	0.44	0.19	0.33
17	0.65	0.07	0.06
18	2.54	0.11	0.14
19	4.02	0.1	0.11
20	4.89	0.07	0.22
21	0.17	0.03	<0.01
22	0.23	0.02	0.01
23	14.29	2.24	1.87
24	16.95	1.01	0.86
25	2.76	0.42	0.88
26	0.26	0.04	0.01
27	0.16	0.01	0.02
28	90.28	10.03	5.02
29	146.79	10.44	5.79
30	2.13	0.04	0.03
31	1.72	0.03	0.03

Appendix A – Test Scripts

Informix

```
#!/bin/ksh

logfile=testifx93.out
bozo=/dev/null

echo "run,query,start(sec),end(sec)" > $logfile

num=1
while (( num < 101 ));do

echo "run number - "$num

# query 1

START="   ,  q1,"`date +%s`
dbaccess  adb_ob4krf  query1.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 2

START="   ,  q2,"`date +%s`
dbaccess  adb_ob4krf  query2.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 3

START="   ,  q3,"`date +%s`
dbaccess  adb_ob4krf  query3.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 4

START="   ,  q4,"`date +%s`
dbaccess  adb_ob4krf  query4.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 5

START="   ,  q5,"`date +%s`
dbaccess  adb_ob4krf  query5.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 6

START="   ,  q6,"`date +%s`
dbaccess  adb_ob4krf  query6.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 7
```

```
START="   ,  q7,"`date +%s`
dbaccess  adb_ob4krf  query7.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 8

START="   ,  q8,"`date +%s`
dbaccess  adb_ob4krf  query8.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 9

START="   ,  q9,"`date +%s`
dbaccess  adb_ob4krf  query9.sql >> $bozo
echo "run "$num$START"   ,     "`date +%s` >> $logfile

# query 10

START="   ,  q10,"`date +%s`
dbaccess  adb_ob4krf  query10.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 11

START="   ,  q11,"`date +%s`
dbaccess  adb_ob4krf  query11.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 12

START="   ,  q12,"`date +%s`
dbaccess  adb_ob4krf  query12.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 13

START="   ,  q13,"`date +%s`
dbaccess  adb_ob4krf  query13.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 14

START="   ,  q14,"`date +%s`
dbaccess  adb_ob4krf  query14.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 15

START="   ,  q15,"`date +%s`
dbaccess  adb_ob4krf  query15.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile



# query 16

START="   ,  q16,"`date +%s`
```

```
dbaccess  adb_ob4krf  query16.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 17

START="   ,  q17,"`date +%s`
dbaccess  adb_ob4krf  query17.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 18

START="   ,  q18,"`date +%s`
dbaccess  adb_ob4krf  query18.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 19

START="   ,  q19,"`date +%s`
dbaccess  adb_ob4krf  query19.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 20

START="   ,  q20,"`date +%s`
dbaccess  adb_ob4krf  query20.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 21

START="   ,  q21,"`date +%s`
dbaccess  adb_ob4krf  query21.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 22

START="   ,  q22,"`date +%s`
dbaccess  adb_ob4krf  query22.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 23

START="   ,  q23,"`date +%s`
dbaccess  adb_ob4krf  query23.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 24

START="   ,  q24,"`date +%s`
dbaccess  adb_ob4krf  query24.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 25

START="   ,  q25,"`date +%s`
dbaccess  adb_ob4krf  query25.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile
```

```
# query 26

START="   ,  q26,"`date +%s`
dbaccess  adb_ob4krf  query26.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 27

START="   ,  q27,"`date +%s`
dbaccess  adb_ob4krf  query27.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 28

START="   ,  q28,"`date +%s`
dbaccess  adb_ob4krf  query28.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 30


START="   ,  q30,"`date +%s`
dbaccess  adb_ob4krf  query30.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile

# query 29

START="   ,  q29,"`date +%s`
dbaccess  adb_ob4krf  query29.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile


# query 31

START="   ,  q31,"`date +%s`
dbaccess  adb_ob4krf  query31.sql >> $bozo
echo "run "$num$START"   ,    "`date +%s` >> $logfile


(( num = num + 1 ))
done

exit
```

PostgreSQL

```
#!/bin/ksh


#PATH=?usr/local/postgres/bin:$PATH
#echo $PATH
#logfile=testpg747.out

logfile=testpg803.200.200.out

bozo=/dev/null

echo "run,query,start,end" > $logfile

num=1
while (( num < 101 ));do

echo "run number - "$num

# query 1

start=",q1,"`date +%s`
psql  -U pguser -d adb_ob7krf -f query1.sql  >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 2

start=",q2,"`date +%s`
psql -U pguser -d adb_ob7krf -f query2.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 3

start=",q3,"`date +%s`
psql -U pguser -d adb_ob7krf -f query3.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 4

start=",q4,"`date +%s`
psql -U pguser -d adb_ob7krf -f query4.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 5

start=",q5,"`date +%s`
psql -U pguser -d adb_ob7krf -f query5.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 6

start=",q6,"`date +%s`
psql -U pguser -d adb_ob7krf -f query6.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile
```

```
# query 7

start=",q7,"`date +%s`
psql -U pguser -d adb_ob7krf -f query7.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 8

start=",q8,"`date +%s`
psql -U pguser -d adb_ob7krf -f query8.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 9

start=",q9,"`date +%s`
psql -U pguser -d adb_ob7krf -f query9.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 10

start=",q10,"`date +%s`
psql -U pguser -d adb_ob7krf -f query10.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 30

start=",q30,"`date +%s`
psql -U pguser -d adb_ob7krf -f query30.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 29

start=",q29,"`date +%s`
psql -U pguser -d adb_ob7krf -f query29.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 11

start=",q11,"`date +%s`
psql -U pguser -d adb_ob7krf -f query11.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 12

start=",q12,"`date +%s`
psql -U pguser -d adb_ob7krf -f query12.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 13

start=",q13,"`date +%s`
psql -U pguser -d adb_ob7krf -f query13.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile
```

```
# query 14

start=",q14,"`date +%s`
psql -U pguser -d adb_ob7krf -f query14.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 15

start=",q15,"`date +%s`
psql -U pguser -d adb_ob7krf -f query15.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 16

start=",q16,"`date +%s`
psql -U pguser -d adb_ob7krf -f query16.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 17

start=",q17,"`date +%s`
psql -U pguser -d adb_ob7krf -f query17.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 18

start=",q18,"`date +%s`
psql -U pguser -d adb_ob7krf -f query18.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 19

start=",q19,"`date +%s`
psql -U pguser -d adb_ob7krf -f query19.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 20

start=",q20,"`date +%s`
psql -U pguser -d adb_ob7krf -f query20.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 21

start=",q21,"`date +%s`
psql -U pguser -d adb_ob7krf -f query21.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 22

start=",q22,"`date +%s`
psql -U pguser -d adb_ob7krf -f query22.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

#----- verify -----
```

```
# query 23

start=",q23,"`date +%s`
psql -U pguser -d adb_ob7krf -f query23.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 24

start=",q24,"`date +%s`
psql -U pguser -d adb_ob7krf -f query24.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 25

start=",q25,"`date +%s`
psql -U pguser -d adb_ob7krf -f query25.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 26

start=",q26,"`date +%s`
psql -U pguser -d adb_ob7krf -f query26.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 27

start=",q27,"`date +%s`
psql -U pguser -d adb_ob7krf -f query27.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 28

start=",q28,"`date +%s`
psql -U pguser -d adb_ob7krf -f query28.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

# query 31

start=",q31,"`date +%s`
psql -U pguser -d adb_ob7krf -f query31.sql >> $bozo
echo "run "$num$start","`date +%s` >> $logfile

(( num = num + 1 ))
done

exit
```

Appendix B – Test Queries

SQL	Informix	PostgreSQL
1	select *from pehpsep where lid='HTNM7' and pe1='H' and pe2='G'and dur='I'and t='1' and s='G' and e='Z' and p='Z' and obstime='2005-04-01';	select * from pehpsep where lid='HTNM7' and pe1='H' and pe2='G' and dur='I' and t='1' and s='G' and e='Z' and p='Z' and obstime='2005-04-01';
2	select *from pedpsep where lid='UNTK1' and pe1='P' and pe2='P' anddur='D' and t='1' and s='Z' and e='Z' and p='Z' and obstime='1996-01';	select * from pedpsep where lid='UNTK1' and pe1='P' and pe2='P' and dur='D' and t='1' and s='Z' and e='Z' and p='Z' and obstime='1996-01-01';
3	select *from pecrsep where lid='HRNM7' and pe1='H' and pe2='G' and dur='I' and t='R' and s='G' and e='Z' and p='Z' and obstime='2001-05-01';	select * from pecrsep where lid='HRNM7' and pe1='H' and pe2='G' and dur='I' and t='R' and s='G' and e='Z' and p='Z' and obstime='2001-05-01';
4	Select bstime, value, shef_qualifier_code, quality_code from pedrsep where lid='SIDM8' and pe1='P' and pe2='P' and dur='P' and t='R' and s='Z' and e='Z' and p='Z' and obstime >= '1993-06-01 00:00:00' and obstime <= '1993-08-01 00:00:00' order by obstime;	select obstime, datavalue, shef_qualifier_code, quality_code from pedrsep where lid='SIDM8' and pe1='P' and pe2='P' and dur='P' and t='R' and s='Z' and e='Z' and p='Z' and obstime >= '1993-06-01 00:00:00' and obstime <= '1993-08-01 00:00:00' order by obstime;
5	select pe1, pe2 from shefpe order by pe1,pe2;	select pe1, pe2 from shefpe order by pe1,pe2;
6	select idur, dur from shefdur order by dur;	select idur, dur from shefdur order by dur;
7	select idur, dur from shefdur order by dur;	select idur, dur from shefdur order by dur;
8	select e from shefex order by e;	select e from shefex order by e;
9	select p, probability from shefprob order by p;	select p, probability from shefprob order by p;
10	select state from state order by state;	select state from state order by state;
11	select wfo_hsa from wfo_hsa order by wfo_hsa;	select wfo_hsa from wfo_hsa order by wfo_hsa;
12	Select count(unique lid) from location where (lid matches '*YANS2*' or name matches '*YANKTON*')and state = 'SD' and wfo = 'FSD';	select count(distinct lid) from location where (lid like '%YANS2%' or name like '%YANKTON%') and state = 'SD' and wfo = 'FSD';
13	select unique lid, name from location where (lid matches '*KCKK1*' or name matches '*KANSAS CITY*') and state = 'KS' and wfo = 'EAX' order by lid;	select distinct lid, name from location where (lid like '%KCKK1%' or name like '%KANSAS CITY%') and state = 'KS' and wfo = 'EAX' order by lid;
14	select name, sbd from location	select name, sbd from location where

	where lid = 'TPAK1' order by sbd desc;	lid = 'TPAK1' order by sbd desc;
15	select * from pecrsep where lid = 'NBDN1' and obstime between '1994-10-15' and '1994-10-20' and pe1 matches 'P' and pe2 matches 'C' and dur matches 'I' and t matches 'R' and s matches 'G' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, obstime desc;	select * from pecrsep where lid = 'NBDN1' and obstime between '1994-10-15' and '1994-10-20' and pe1 like 'P' and pe2 like 'C' and dur like 'I' and t like 'R' and s like 'G' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, obstime desc;
16	select * from pedrsep where lid = 'MLFK1' and obstime between '2001-08-01 00:00:00' and '2001-08-04 00:00:00' and pe1 matches 'Q' and pe2 matches 'I' and dur matches 'D' and t matches 'R' and s matches 'Z' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, obstime desc;	select * from pedrsep where lid = 'MLFK1' and obstime between '2001-08-01 00:00:00' and '2001-08-04 00:00:00' and pe1 like 'Q' and pe2 like 'I' and dur like 'D' and t like 'R' and s like 'Z' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, obstime desc;
17	select * from peoosep where lid = ‘DLGM8” and obstime between “2005-01-01 00:00:00” and “2005-02-01 00:00:00” and pe1 matches “S” and pe2 matches “W” and dur matches “I” and t matches “P” and s matches “Z” and e matches “Z” and p matches “Z” order by pe1, pe2, dur, obstime desc;	select * from peoosep where lid = 'DLGM8' and obstime between '2005-01-01 00:00:00' and '2005-02-01 00:00:00' and pe1 like 'S' and pe2 like 'W' and dur like 'I' and t like 'P' and s like 'Z' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, obstime desc
18	select * from pedfsep where lid = 'BILM8' and basistime between '2004-05-21 00:00:00' and '2004-05-31 00:00:00' and pe1 matches 'H' and pe2 matches 'G' and dur matches 'I' and t matches 'F' and s matches 'F' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, basistime desc;	select * from pedfsep where lid = 'BILM8' and basistime between '2004-05-21 00:00:00' and '2004-05-31 00:00:00' and pe1 like 'H' and pe2 like 'G' and dur like 'I' and t like 'F' and s like 'F' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, basistime desc;
19	select * from pedcsep where lid = 'MNC133' and basistime between '2003-08-22 00:00:00' and '2003-08-26 00:00:00' and pe1 matches 'P' and pe2 matches 'P' and dur matches 'T' and t matches 'C' and s matches 'F' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, basistime desc;	select * from pedcsep where lid = 'MNC133' and basistime between '2003-08-22 00:00:00' and '2003-08-26 00:00:00' and pe1 like 'P' and pe2 like 'P' and dur like 'T' and t like 'C' and s like 'F' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, basistime desc;
20	select * from pehpsep where lid = 'NVDM7' and obstime between '1997-09-01' and '1998-01-01' and pe1 matches 'H' and pe2 matches 'G' and dur matches 'I' and t matches '1' and s matches	select * from pehpsep where lid = 'NVDM7' and obstime between '1997-09-01' and '1998-01-01' and pe1 like 'H' and pe2 like 'G' and dur like 'I' and t like '1' and s like 'G' and e like 'Z' and p like 'Z'

	'G' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, obstime desc;	order by pe1,pe2, dur, obstime desc;
21	select * from pedpsep where lid = 'UNTK1' and obstime between '1997-01' and '1998-01' and pe1 matches 'P' and pe2 matches 'P' and dur matches 'D' and matches '1' and s matches 'Z' and e matches 'Z' and p matches 'Z' order by pe1,pe2, dur, obstime desc;	select * from pedpsep where lid = 'UNTK1' and obstime between '1997-01-01' and '1998-01-01' and pe1 like 'P' and pe2 like 'P' and dur like 'D' and t like '1' and s like 'Z' and e like 'Z' and p like 'Z' order by pe1,pe2, dur, obstime desc;
22	select * from vfyruninfo where lid in ('KCDM7') ORDER BY lid,pe1,pe2,senrk ASC;	select * from vfyruninfo where lid in ('KCDM7') ORDER BY lid,pe1,pe2,senrk ASC;
23	select * from vfypairs where lid='KCDM7' AND (pe1 || pe2)='HG' AND (fcst_t || fcst_s) in ('FF') AND validtime between '2005-04-01 00:00:00' and '2005-04-30 23:59:59' AND (validtime - basistime) > 0 UNITS HOUR AND (validtime - basistime) <= 72 UNITS HOUR ;	select * from vfypairs where lid='KCDM7' AND (pe1 || pe2)='HG' AND (fcst_t || fcst_s) in ('FF') AND validtime between '2005-04-01 00:00:00' AND '2005-04-30 23:59:59' AND EXTRACT(HOUR FROM validtime - basistime) > 0 AND EXTRACT(HOUR FROM validtime - basistime) <= 72;
24	select * from pedfsep where lid in ('KCDM7') AND pe1 || pe2 in ('HG') AND t || s in ('FF') AND e in ('Z') AND validtime between '2005-01-01 00:00:00' and '2005-01-31 23:59:59' AND value > -999.0;	select * from pedfsep where lid in ('KCDM7') AND pe1 || pe2 in ('HG') AND t || s in ('FF') AND e in ('Z') AND validtime between '2005-01-01 00:00:00' and '2005-01-31 23:59:59' AND datavalue > -999.0;
25	select * from pecrsep where lid in ('KCDM7') AND pe1 || pe2 in ('HG') AND e in ('Z') AND obstime between '2004-12-31' and '2005-02-01';	select * from pecrsep where lid in ('KCDM7') AND pe1 || pe2 in ('HG') AND e in ('Z') AND obstime between '2004-12-31' and '2005-02-01';
26	select * from riverstat WHERE lid='KCDM7' AND sed is NULL;	select * from rivercrit WHERE lid='KCDM7' AND pe1 || pe2 IN ('HG','HT') ORDER BY vdtime DESC;
27	select * from location where lid='KCDM7' and sed is NULL;	select * from location where lid='KCDM7' and sed is NULL;
28	select * from pecrsep where lid='KCDM7' and pe1='H' and pe2='G';	select * from pecrsep where lid='KCDM7' and pe1='H' and pe2='G';
29	select * from pecrsep where lid='KCDM7' and pe1='H' and pe2='G' order by obstime;	select * from pecrsep where lid='KCDM7' and pe1='H' and pe2='G' order by obstime;
30	Select lid, pe1, obstim, value from pedrsep where lid='GTFM8' and pe1='T' and pe2='A' and dur='I' and e in ('X','N');	select lid,pe1,pe2,e,obstime,datavalue from pedrsep where lid='GTFM8' and pe1='T' and pe2='A' and dur='I' and e in ('X','N');

31	Select lid, pe1,pe2, e,obstime, value from pedrsep where lid='GTFM8' and pe1='T' and pe2='A' and dur='I' and e in ('X','N') order by obstime;	select lid, pe1, pe2, e, obstime, datavalue from pedrsep where lid='GTFM8' and pe1='T' and pe2='A' and dur='I' and e in ('X','N') order by obstime;

Appendix C – Test Results Tables

PostgreSQL DB RHE 3.0 ax2-nhdr build ob7

Query	Table	Total number of rows in table	Number of rows retrieved	Ver. 7.4.7 Average runtime (seconds)	Ver. 8.0.3 Average Runtime (seconds)
1	pehpsep	35,668	1	0.02	0.03
2	pedpsep	84	1	0.02	0.01
3	pecrsep	11,336,999	1	0.31	0.64
4	pedrsep	15,610,991	44	0.06	0.11
5	shefpe	218	218	0.02	0.04
6	shefdur	23	23	<0.01	0.03
7	shefts	206	206	0.01	0.02
8	shefex	21	21	0.04	<0.01
9	shefprob	31	31	0.01	0.01
10	state	109	109	0.02	0.02
11	wfo_hsa	125	125	0.03	0.03
12	location	15,490	1	0.07	0.05
13	location	15.490	1	0.06	0.03
14	location	15,490	1	0.03	0.01
15	pecrsep	11,336,999	6	0.09	0.08
16	pedrsep	15,610,991	2	0.19	0.33
17	peoosep	1,572.005	31	0.07	0.06
18	pedfsep	10,185,188	200	0.11	0.14
19	pedcsep	1,805,396	8	0.1	0.11
20	pehpsep	35,668	123	0.07	0.22
21	pedpsep	84	1	0.03	<0.01
22	vfyruninfo	1,132	3	0.02	0.01
23	vfypairs	1,104,303	360	2.24	1.87
24	pedfsep	10,185,188	945	1.01	0.86
25	pecrsep	11,336,999	67	0.42	0.88
26	rivercrit	715	1	0.04	0.01
27	location	15,490	1	0.01	0.02
28	pecrsep	11,336,999	12,533	10.03	5.02
29	pecrsep	11,336,999	12,533	10.44	5.79
30	pedrsep	15,610,991	200	0.04	0.03
31	pedrsep	15,610,991	200	0.03	0.03

Informix DB 9.30.UC1 RH7.2 ax-KRF build ob4

Query	Table	Total number of rows in table	Number of rows retrieved	Average runtime (seconds)
1	pehpsep	35,668	1	0.2
2	pedpsep	84	1	0.26
3	pecrsep	11,360,357	1	1.46
4	pedrsep	15,634,408	44	4.71
5	shefpe	218	218	0.18
6	shefdur	23	23	0.07
7	shefts	206	206	0.09
8	shefex	21	21	0.16
9	shefprob	31	31	0.08
10	state	109	109	0.09
11	wfo_has	125	125	0.03
12	location	15,490	1	3.8
13	location	15.49-	1	3.61
14	location	15,490	1	0.14
15	pecrsep	11,360,357	6	1.99
16	pedrsep	15,634,408	2	0.44
17	peoosep	1,602,110	31	0.65
18	pedfsep	10,232,326	200	2.54
19	pedcsep	1,826,041	8	4.02
20	pehpsep	35,668	123	4.89
21	pedpsep	84	1	0.17
22	vfyruninfo	1,132	3	0.23
23	vfypairs	1,144,465	360	14.29
24	pedfsep	10.232.326	945	16.95
25	pecrsep	11,360.357	67	2.76
26	riverstat	1031	1	0.26
27	location	15,490	1	0.16
28	pecrsep	11,360,357	12,533	90.28
29	pecrsep	11,360,357	12,533	146.79
30	pedrsep	15,634,408	200	2.13
31	pedrsep	15,634,408	200	1.72

Appendix D – Informix onconfig file

```
#**************************************************************************
#
#                         INFORMIX SOFTWARE, INC.
#
#  Title:      onconfig.std
#  Description: Informix Dynamic Server Configuration Parameters
#
#**************************************************************************

# Root Dbspace Configuration

ROOTNAME        rootdbs         # Root dbspace name
ROOTPATH        /opt/informix/links/rootdbs # Path to root dbspace link
ROOTOFFSET      0               # Offset of root dbspace into device (Kbytes)
ROOTSIZE        450000          # Size of root dbspace (Kbytes)

# Disk Mirroring Configuration Parameters

MIRROR          0               # Mirroring flag (Yes = 1, No = 0)
MIRRORPATH                      # Path for device containing mirrored root
MIRROROFFSET    0               # Offset into mirrored device (Kbytes)

# Physical Log Configuration

PHYSDBS         rootdbs         # Location (dbspace) of physical log
PHYSFILE        30000           # Physical log file size (Kbytes)

# Logical Log Configuration

LOGFILES        31              # Number of logical log files
LOGSIZE         3000            # Logical log size (Kbytes)

# Diagnostics

MSGPATH         /opt/informix/online.log # System message log file path
CONSOLE         /dev/console    # System console message path
ALARMPROGRAM    # Alarm program path
SYSALARMPROGRAM /opt/informix/etc/evidence.sh # System alarm program path
TBLSPACE_STATS  1               # Maintain tblspace statistics

# System Archive Tape Device

TAPEDEV         /opt/informix/links/tapedev # Tape device path
TAPEBLK         1024            # Tape block size (Kbytes)
TAPESIZE        40960000        # Maximum amount of data to put on tape (Kbytes)

# Log Archive Tape Device

LTAPEDEV        /dev/null       # Log tape device path
LTAPEBLK        1024            # Log tape block size (Kbytes)
LTAPESIZE       40960000        # Max amount of data to put on log tape (Kbytes)

# Optical

STAGEBLOB                       # Informix Dynamic Server staging area

# System Configuration

SERVERNUM       108             # Unique id corresponding to a OnLine instance
DBSERVERNAME    adbs            # Name of default database server
DBSERVERALIASES                 # List of alternate dbservernames
DEADLOCK_TIMEOUT 60              # Max time to wait of lock in distributed env.
RESIDENT        0               # Forced residency flag (Yes = 1, No = 0)

MULTIPROCESSOR  0               # 0 for single-processor, 1 for multi-processor
NUMCPUVPS       1               # Number of user (cpu) vps
SINGLE_CPU_VP   0               # If non-zero, limit number of cpu vps to one
```

```
NOAGE           0               # Process aging
AFF_SPROC       0               # Affinity start processor
AFF_NPROCS      0               # Affinity number of processors

# Shared Memory Parameters

LOCKS           20000           # Maximum number of locks
BUFFERS         2000            # Maximum number of shared buffers
NUMAIOVPS       2               # Number of IO vps
PHYSBUFF        32              # Physical log buffer size (Kbytes)
LOGBUFF         32              # Logical log buffer size (Kbytes)
CLEANERS        24              # Number of buffer cleaner processes
SHMBASE         0x10000000        # Shared memory base address
SHMVIRTSIZE     8000            # initial virtual shared memory segment size
SHMADD          8192            # Size of new shared memory segments (Kbytes)
SHMTOTAL        0               # Total shared memory (Kbytes). 0=>unlimited
CKPTINTVL       300             # Check point interval (in sec)
LRUS            8               # Number of LRU queues
LRU_MAX_DIRTY   60              # LRU percent dirty begin cleaning limit
LRU_MIN_DIRTY   50              # LRU percent dirty end cleaning limit
TXTIMEOUT       0x12c             # Transaction timeout (in sec)
STACKSIZE       32              # Stack size (Kbytes)

# System Page Size
# BUFFSIZE - OnLine no longer supports this configuration parameter.
#            To determine the page size used by OnLine on your platform
#            see the last line of output from the command, 'onstat -b'.


# Recovery Variables
# OFF_RECVRY_THREADS:
# Number of parallel worker threads during fast recovery or an offline restore.
# ON_RECVRY_THREADS:
# Number of parallel worker threads during an online restore.

OFF_RECVRY_THREADS 10              # Default number of offline worker threads
ON_RECVRY_THREADS 1               # Default number of online worker threads

# Data Replication Variables
DRINTERVAL      30              # DR max time between DR buffer flushes (in sec)
DRTIMEOUT       30              # DR network timeout (in sec)
DRLOSTFOUND     /usr/informix/etc/dr.lostfound # DR lost+found file path

# CDR Variables
CDR_EVALTHREADS 1,2             # evaluator threads (per-cpu-vp,additional)
CDR_DSLOCKWAIT  5               # DS lockwait timeout (seconds)
CDR_QUEUEMEM    4096            # Maximum amount of memory for any CDR queue (Kbytes)
CDR_NIFCOMPRESS 0               # Link level compression (-1 never, 0 none, 9 max)
CDR_SERIAL      0,0             # Serial Column Sequence
CDR_DBSPACE                     # dbspace for syscdr database
CDR_QHDR_DBSPACE                 # CDR queue dbspace (default same as catalog)
CDR_QDATA_SBSPACE                # CDR queue smart blob space
CDR_QDATA_SBFLAGS 0              # Log/no-log (default no log)


# Backup/Restore variables
BAR_ACT_LOG     /opt/informix/bar_act.log # ON-Bar Log file - not in /tmp please
BAR_DEBUG_LOG   /opt/informix/bar_dbug.log
                                # ON-Bar Debug Log - not in /tmp please
BAR_MAX_BACKUP  0
BAR_RETRY       1
BAR_NB_XPORT_COUNT 10
BAR_XFER_BUF_SIZE 31
RESTARTABLE_RESTORE on
BAR_PROGRESS_FREQ 0

# Informix Storage Manager variables
ISM_DATA_POOL   ISMData
ISM_LOG_POOL    ISMLogs

# Read Ahead Variables
```

```
RA_PAGES        4               # Number of pages to attempt to read ahead
RA_THRESHOLD    2               # Number of pages left before next group

# DBSPACETEMP:
# OnLine equivalent of DBTEMP for SE. This is the list of dbspaces
# that the OnLine SQL Engine will use to create temp tables etc.
# If specified it must be a colon separated list of dbspaces that exist
# when the OnLine system is brought online.  If not specified, or if
# all dbspaces specified are invalid, various ad hoc queries will create
# temporary files in /tmp instead.

DBSPACETEMP                     # Default temp dbspaces

# DUMP*:
# The following parameters control the type of diagnostics information which
# is preserved when an unanticipated error condition (assertion failure) occurs
# during OnLine operations.
# For DUMPSHMEM, DUMPGCORE and DUMPCORE 1 means Yes, 0 means No.

DUMPDIR         /tmp            # Preserve diagnostics in this directory
DUMPSHMEM       1               # Dump a copy of shared memory
DUMPGCORE       0               # Dump a core image using 'gcore'
DUMPCORE        0               # Dump a core image (Warning:this aborts OnLine)
DUMPCNT         1               # Number of shared memory or gcore dumps for
                               # a single user's session

FILLFACTOR      90              # Fill factor for building indexes

# method for OnLine to use when determining current time
USEOSTIME       0               # 0: use internal time(fast), 1: get time from OS(slow)

# Parallel Database Queries (pdq)
MAX_PDQPRIORITY 100             # Maximum allowed pdqpriority
DS_MAX_QUERIES  2               # Maximum number of decision support queries
DS_TOTAL_MEMORY 256             # Decision support memory (Kbytes)
DS_MAX_SCANS    1048576         # Maximum number of decision support scans
DATASKIP        off             # List of dbspaces to skip

# OPTCOMPIND
# 0 => Nested loop joins will be preferred (where
#      possible) over sortmerge joins and hash joins.
# 1 => If the transaction isolation mode is not
#      "repeatable read", optimizer behaves as in (2)
#      below.  Otherwise it behaves as in (0) above.
# 2 => Use costs regardless of the transaction isolation
#      mode.  Nested loop joins are not necessarily
#      preferred.  Optimizer bases its decision purely
#      on costs.
OPTCOMPIND      2               # To hint the optimizer

DIRECTIVES      1               # Optimizer DIRECTIVES ON (1/Default) or OFF (0)

ONDBSPACEDOWN   2               # Dbspace down option: 0 = CONTINUE, 1 = ABORT, 2 = WAIT
OPCACHEMAX      0               # Maximum optical cache size (Kbytes)

# HETERO_COMMIT (Gateway participation in distributed transactions)
# 1 => Heterogeneous Commit is enabled
# 0 (or any other value) => Heterogeneous Commit is disabled
HETERO_COMMIT   0

SBSPACENAME                     # Default smartblob space name - this is where blobs
                      # go if no sbspace is specified when the smartblob is
                      # created. It is also used by some datablades as
                      # the location to put their smartblobs.
SYSSBSPACENAME                  # Default smartblob space for use by the Informix
                      # Server. This is used primarily for Informix Server
                      # system statistics collection.

BLOCKTIMEOUT    3600            # Default timeout for system block

# Optimization goal: -1 = ALL_ROWS(Default), 0 = FIRST_ROWS
```

```
OPT_GOAL        -1

ALLOW_NEWLINE   0               # embedded newlines(Yes = 1, No = 0 or anything but 1)

#
# The following are default settings for enabling Java in the database.
# Replace all occurrences of /usr/informix with the value of $INFORMIXDIR.

#VPCLASS        jvp,num=1       # Number of JVPs to start with

JVPJAVAHOME     /usr/informix/extend/krakatoa/jre
                                # JRE installation root directory
JVPHOME         /usr/informix/extend/krakatoa # Krakatoa installation directory

JVPPROPFILE     /usr/informix/extend/krakatoa/.jvpprops # JVP property file

JDKVERSION      1.3             # JDK version supported by this server

# The path to the JRE libraries relative to JVPJAVAHOME
JVPJAVALIB      /lib/i386/

# The JRE libraries to use for the Java VM

JVPJAVAVM       hpi:server:verify:java:net:zip:jpeg

# use JVPARGS to change Java VM configuration
#To display jni call
#JVPARGS       -verbose:jni

# Classpath to use upon Java VM start-up (use _g version for debugging)

#JVPCLASSPATH
/usr/informix/extend/krakatoa/krakatoa_g.jar:/usr/informix/extend/krakatoa/jdbc_g.jar
JVPCLASSPATH    /usr/informix/extend/krakatoa/krakatoa.jar:/usr/informix/extend/krakatoa/jdbc.jar
```

Appendix E – PostgreSQL V8.0.3 Configuration

name	setting
add_missing_from	on
archive_command	unset
australian_timezones	off
authentication_timeout	60
bgwriter_delay	200
bgwriter_maxpages	100
bgwriter_percent	1
block_size	8192
check_function_bodies	on
checkpoint_segments	3
checkpoint_timeout	300
checkpoint_warning	30
client_encoding	UNICODE
client_min_messages	notice
commit_delay	0
commit_siblings	5
config_file	/data/pgdata/8.0.3/postgresql.conf
cpu_index_tuple_cost	0.001
cpu_operator_cost	0.0025
cpu_tuple_cost	0.01
custom_variable_classes	unset
data_directory	/data/pgdata/8.0.3
DateStyle	ISO, MDY
db_user_namespace	off
deadlock_timeout	1000
debug_pretty_print	off
debug_print_parse	off
debug_print_plan	off
debug_print_rewritten	off
debug_shared_buffers	0
default_statistics_target	10
default_tablespace	unset
default_transaction_isolation	read committed
default_transaction_read_only	off
default_with_oids	on
dynamic_library_path	$libdir
effective_cache_size	1000
enable_hashagg	on
enable_hashjoin	on
enable_indexscan	on
enable_mergejoin	on
enable_nestloop	on
enable_seqscan	on
enable_sort	on
enable_tidscan	on
explain_pretty_print	on
external_pid_file	unset
extra_float_digits	0
from_collapse_limit	8
fsync	on
geqo	on
geqo_effort	5
geqo_generations	0
geqo_pool_size	0
geqo_selection_bias	2
geqo_threshold	12
hba_file	/data/pgdata/8.0.3/pg_hba.conf
ident_file	/data/pgdata/8.0.3/pg_ident.conf
integer_datetimes	off
join_collapse_limit	8
krb_server_keyfile	FILE:/etc/sysconfig/pgsql/krb5.keytab
lc_collate	en_US.UTF-8
lc_ctype	en_US.UTF-8
lc_messages	en_US.UTF-8

```
lc_monetary                    | en_US.UTF-8
lc_numeric                     | en_US.UTF-8
lc_time                        | en_US.UTF-8
listen_addresses               | localhost
log_connections                | off
log_destination                | stderr
log_directory                  | pg_log
log_disconnections             | off
log_duration                   | off
log_error_verbosity            | default
log_executor_stats             | off
log_filename                   | postgresql-%a.log
log_hostname                   | off
log_line_prefix                | unset
log_min_duration_statement     | -1
log_min_error_statement        | panic
log_min_messages               | notice
log_parser_stats               | off
log_planner_stats              | off
log_rotation_age               | 1440
log_rotation_size              | 0
log_statement                  | none
log_statement_stats            | off
log_truncate_on_rotation       | on
maintenance_work_mem           | 16384
max_connections                | 100
max_files_per_process          | 1000
max_fsm_pages                  | 20000
max_fsm_relations              | 1000
max_function_args              | 32
max_identifier_length          | 63
max_index_keys                 | 32
max_locks_per_transaction      | 64
max_stack_depth                | 2048
password_encryption            | on
port                           | 5432
pre_auth_delay                 | 0
preload_libraries              | unset
random_page_cost               | 4
redirect_stderr                | on
regex_flavor                   | advanced
rendezvous_name                | unset
search_path                    | $user,public
server_encoding                | UNICODE
server_version                 | 8.0.3
shared_buffers                 | 1000
silent_mode                    | off
sql_inheritance                | on
ssl                            | off
statement_timeout              | 0
stats_block_level              | off
stats_command_string           | off
stats_reset_on_server_start    | on
stats_row_level                | off
stats_start_collector          | on
superuser_reserved_connections | 2
syslog_facility                | LOCAL0
syslog_ident                   | postgres
TimeZone                       | GMT
trace_notify                   | off
transaction_isolation          | read committed
transaction_read_only          | off
transform_null_equals          | off
unix_socket_directory          | unset
unix_socket_group              | unset
unix_socket_permissions        | 511
stats_start_collector          | on
superuser_reserved_connections | 2
syslog_facility                | LOCAL0
syslog_ident                   | postgres
TimeZone                       | GMT
```

```
trace_notify                    | off
transaction_isolation           | read committed
transaction_read_only           | off
transform_null_equals           | off
unix_socket_directory           | unset
unix_socket_group               | unset
unix_socket_permissions         | 511
vacuum_cost_delay               | 0
vacuum_cost_limit               | 200
vacuum_cost_page_dirty          | 20
vacuum_cost_page_hit            | 1
vacuum_cost_page_miss           | 10
wal_buffers                     | 8
wal_sync_method                 | fdatasync
work_mem                        | 1024
zero_damaged_pages              | off
```

Appendix F – PostgreSQL V7.4.7 Configuration

name	setting
add_missing_from	on
australian_timezones	off
authentication_timeout	60
check_function_bodies	on
checkpoint_segments	3
checkpoint_timeout	300
checkpoint_warning	30
client_encoding	SQL_ASCII
client_min_messages	notice
commit_delay	0
commit_siblings	5
cpu_index_tuple_cost	0.001
cpu_operator_cost	0.0025
cpu_tuple_cost	0.01
DateStyle	ISO, MDY
db_user_namespace	off
deadlock_timeout	1000
debug_pretty_print	off
debug_print_parse	off
debug_print_plan	off
debug_print_rewritten	off
default_statistics_target	10
default_transaction_isolation	read committed
default_transaction_read_only	off
dynamic_library_path	$libdir
effective_cache_size	1000
enable_hashagg	on
enable_hashjoin	on
enable_indexscan	on
enable_mergejoin	on
enable_nestloop	on
enable_seqscan	on
enable_sort	on
enable_tidscan	on
explain_pretty_print	on
extra_float_digits	0
from_collapse_limit	8
fsync	on
geqo	on
geqo_effort	1
geqo_generations	0
geqo_pool_size	0
geqo_pool_size	0
geqo_selection_bias	2
geqo_threshold	11
join_collapse_limit	8
krb_server_keyfile	unset
lc_collate	en_US.UTF-8
lc_ctype	en_US.UTF-8
lc_messages	en_US.UTF-8
lc_monetary	en_US.UTF-8
lc_numeric	en_US.UTF-8
lc_time	en_US.UTF-8
log_connections	off
log_duration	off
log_error_verbosity	default
log_executor_stats	off
log_hostname	off
log_min_duration_statement	-1
log_min_error_statement	panic
log_min_messages	notice
log_parser_stats	off
log_pid	off
log_planner_stats	off

```
log_source_port                | off
log_statement                  | off
log_statement_stats            | off
log_timestamp                  | off
max_connections                | 100
max_expr_depth                 | 10000
max_files_per_process          | 1000
max_fsm_pages                  | 20000
max_fsm_relations              | 1000
max_locks_per_transaction      | 64
password_encryption            | on
port                           | 5432
pre_auth_delay                 | 0
preload_libraries              | unset
random_page_cost               | 4
regex_flavor                   | advanced
rendezvous_name                | unset
search_path                    | $user,public
server_encoding                | SQL_ASCII
server_version                 | 7.4.7
shared_buffers                 | 1000
silent_mode                    | off
sort_mem                       | 1024
sql_inheritance                | on
ssl                            | off
statement_timeout              | 0
stats_block_level              | off
stats_command_string           | off
stats_reset_on_server_start    | on
stats_row_level                | off
stats_start_collector          | on
superuser_reserved_connections | 2
syslog                         | 0
syslog_facility                | LOCAL0
syslog_ident                   | postgres
tcpip_socket                   | off
TimeZone                       | unknown
trace_notify                   | off
transaction_isolation          | read committed
transaction_read_only          | off
transform_null_equals          | off
unix_socket_directory          | unset
unix_socket_group              | unset
unix_socket_permissions        | 511
vacuum_mem                     | 8192
virtual_host                   | unset
wal_buffers                    | 8
wal_debug                      | 0
wal_sync_method                | fdatasync
zero_damaged_pages             | off
```