Common Database Problems Common Database Solutions

Mike Furgal

Director – Database Services

BravePoint

mfurgal@bravepoint.com

BRAVEPOINT

Introduction- Mike Furgal

- Progress Employee from 1989 to 1998
- BravePoint Consultant from 1999 to 2000
- Progress Employee (again) 2001 to 2012
- BravePoint Consultant 2012 to now
- Progress/OpenEdge Database Expert

Introduction - BravePoint

- **FACTS**
  - The Largest Progress consulting group in the world
  - Founded in 1987
  - Formerly called USI
  - Sponsor of the Bunker Tests
  - The Managed Database Team has 5 former Progress Employees
  - Dan Foreman, John Harlow
  - Purveyor of many OpenEdge related books

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

Disaster Area

Disaster Area

Case Study 1

- A large distribution center had a power failure. When the power came back on the machine booted but the database did not start

```
[2013/08/02@09:04:07.626-0500] P-7188      T-507976656 I          : (43)    **
Cannot find or open file /agility/prod/prod_db/platte_11.d5, errno = 2.

[2013/08/02@09:09:57.111-0500] P-8609      T--1155527728 I         : (451)
prostrct list session begin for root on /dev/pts/0.
[2013/08/02@09:09:57.116-0500] P-8609      T--1155527728 I         : (12475)
Unable to get file status for extent /agility/prod/prod_db/platte_11.d5
[2013/08/02@09:09:57.116-0500] P-8609      T--1155527728 I         : (334)
prostrct list session end.
```

Specifics

- Database was 80 GB
- Last good backup was 1 week old
- Not running After Imaging
- Platform was Linux

WHAT WOULD YOU DO?

Approach

- **Made a copy of the existing database incase we made a mistake**
- **Used PROSTRCT LIST to determine which files were missing**
  - We were lucky that the missing file was part of a storage area that only held indexes
- **Tools Available**
  - PROSTRCT UNLOCK
  - PROSTRCT BUILDDB

Solution

- Restored the missing extent from the week old backup and ran PROSTRCT UNLOCK
- Rebuilt the indexes
  - # proutil db –C idxbuild all

  BUT…….
- Index rebuild failed due to finding back blocks in the storage area where the records were stored

NOW WHAT?

Back to the Beginning

- **Copied the backed up database to start over.**
  - Good thing we had copied all the files in the first place
- **Add the missing extent**
- **Truncate the BI and do a DBRPR scan**
  - Fix bad blocks
  - Fix bad records

Dump and Load

- After all the corruption was removed it was time to dump and load
- Need to do an ASCII Dump to dump around some bad records

Lessons Learned

- This Database was important to this customer, hence they wanted it back when it got corrupted.
- They need to treat the Database better
  - Daily Backups
  - After Imaging
- A good DR plan saves a lot of heartache

Next Step

Next Steps

- **Implement a good Disaster Recover plan which includes**
  - Frequent backups
  - After Imaging implemented
- **Test the Disaster Recover Plan**
  - Annually
- **Disaster Recover Plan needs to be on Paper**
  - Can’t be just on the computer
  - Need a backup plan incase the DR plan fails

Disaster Area

Disaster Area

- **Case Study 2**
  - A brand name bank had SAN corruption. This prevented Crash Recovery from completing. They had a Hot Standby machine and database using OE Replication.

Specifics

- Had a local backup and local AI files, but the backup would not restore
- Previous backup was not available
- Replica Database was up to date
- Platform was Windows
- Database size 50 GB

What's the Problem

- **Customer refused to fail-over**
  - They never tested running on the fail-over machine. Had little confidence that the application would run in the fail-over environment.
  - Customer worried about the time it takes to fail-back once failed over.

Making Matters Worse

- Copying the DR database to the production machine is measured in days
- Options presented to Management included FORCED ACCESS to the Database

What Next

- Forced into the database.
- Index Rebuild DOES NOT fix the database
- Dump and Load DOES NOT fix the database

Lesson Learned

- **Have confidence in your Disaster Recovery Plan**
  - There is no sense of having one if you are never going to use it
- **Be Careful of the “QUICK FIX”**
  - Non technical people will ALWAYS choose the fastest approach to the solution without understanding the consequences

Next Steps

Next Steps

- Worked with the customer to do a fail-over test.
- Made the fail-over testing an annual event

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

Performance Problems

- **Case Study 3**
  - A Large OpenEdge deployment just purchased their biggest competitor. Their workload is expected to double. Can their environment handle it?
  - Specifics
    - 2 Databases each 1 TB
    - 3,000+ Users
    - Nightly Processing 1 MM “transactions” takes 4 hours, expected to grow to 2 MM
    - Platform Solaris

Tune Tune Tune

- Database working set in memory
- Using Alternate Buffer Pool
- Using LRUSkips
- Spinlocks in use
- BI Cluster Size tuned
- No Disk Bottleneck

- All the usual suspects accounted for

Problem Still Exists

WHAT WOULD YOU DO?

Where's the problem?

- **Processing 2 MM rows takes to long.**
- **Thinking out of the box**
  - Can we dump the data for this “transaction” to a different machine, processes it there and load it back in?
  - Can we change the application so it doesn't do the 2 MM rows in batch, but does them over time?
  - Can we separate the tables in this “transaction” into their own database?

Symptoms

- **During the batch run, collected many statistics from PROMON**
- **Found MANY hundreds of blocked clients**
  - Large percentage blocked on a DB Block share lock on a specific block

SEEMS LIKE A CLUE

Solution

- Noticed that the table with the most creates had many indexes
  - -tablerangesize and _TableStatus
- Noticed that many of the indexes were never used
  - _indexrangesize and _IndexStatus
- Deactivating the unused indexes allowed the processing time to decrease dramatically

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

YOU GET TO PIC THE NEXT MIGRATION PICTURE

Migrations and Upgrades

- **Case Study 4**
  - A large warehouse distribution center is migrating from an older machine to a newer machine. They are 24x7x365, so downtime is a minimum.
- **Specifics**
  - Not changing platforms (AIX -> AIX)
  - DB size is 500 GB, it takes 6 hours to copy it the new machine

Constraints

- Blocked out a 2 hour window for the data migration

- Do you need any more constraints than this?

Approach

- **Build a test environment and TEST TEST TEST**
- **Migrate:**
  - Application Files
  - System Files
  - Other files (in this case, custom terminfo)

TEST TEST TEST

BRAVEPOINT

Special Sauce

- Use After Imaging to keep the soon to be production database in synch.
- During the cut-over period only need to transfer last AI file and apply it
- This fits into the 2 hour downtime window with ease

Application Upgrades

- **Case Study 5**
  - A customer has replication in place using AI files. They are an end user of an Application Partner who does frequent updates of the application. The Application Partner is aware of the DR machine, and updates the application on the DR machine. Sometimes this causes problems
- **Specifics**
  - Windows Application

The Problem

The application support personnel doing the upgrade follows a a script which at times includes not only updating the Application, but also connecting to the database

The Problem

Once the database is connected to, crash recovery is performed and no other AI files can be applied

So…..

- When we can’t apply any more AI files, we need to rebaseline the database. The database size is 80GB and the replica is on a WAN, so rebaselining takes 1-2 days

Solution #1

- **Use the oplock and opunlock commands.**
  - rfutil dbname –C roll forward oplock –a file.a1
  - rfutil dbname –C roll forward opunlock

Solution #1

However, this prevented the support engineer from doing the upgrade properly as they didn’t have control over when/how it connected to the database

Solution #2

- **We provided a secondary database that is a backup of the replica database on the DR machine**
  - We used probkup with the –norecovery switch to make sure future AI files can still be applied

Solution #2

- **The support engineer was able to complete the upgrade using this database copy**
- **The Database Changes from the upgrade are THROW AWAY**
  - These changes will migrate from the PRODUCTION Database to the REPLICA Database via Replication

Summary

- **These are examples of some common and maybe not so common Database Problems**
- **Don’t assume things can’t go wrong**
- **Having a plan is not going enough**
  - Testing the plan and having confidence is required
- **If all else fails, seek professional help**

Questions

THANK YOU FOR YOUR TIME