Common Database Problems
Common Database Solutions

Mike Furgal
PROGRESS Bravepoint – Database Services

Introduction

Mike Furgal

- Progress Employee since 1989
- Developer of the OpenEdge database
- Joined Bravepoint in 2012
- Heads up Database Services
  - Including Managed Database Services

Bravepoint

- Largest Progress/OpenEdge consulting firm
- Founded in 1987
- Purchased by Progress in April 2014
- Specializes in all things OpenEdge
  - Database Services
  - Programming
  - QAD
- Pro2SQL
  - Real-time Replication to SQL Target

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

APJ
PROGRESS SPARK

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.

APJ
PROGRESS SPARK

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 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

APJ
PROGRESS SPARK

APJ
PROGRESS SPARK

Disaster Area

- Case Study 2
  - A brand name US 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 200 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

APJ
PROGRESS SPARK

What Next

- Forced into the database – This skips Crash Recovery
- 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

APJ
PROGRESS SPARK

Next Steps

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

APJ
PROGRESS SPARK

Disaster Area

- Case Study 3
  - A Large school district needs to get their reports cards out to 30,000+ students. They discovered they had corruption in the database because backups stopped working for about a week

Specifics

- 10.2B05 Windows 64bit OpenEdge
- Last good backup is 1 week old
- All report card data for 30,000+ students entered since that last good backup
- After Imaging is turned on, but AI file retention was less than 1 week
- Database is about 300 GB
- They have the 1 week old backup restored to a different location

WHAT WOULD YOU DO?

Approach

- We had 2 plans

- Plan A – Get the corruption out of the live database
  - Use any and all tools to remove the corruption

- Plan B – Revert back to the week old database
  - See if we can take all the report card data from the live database and import it into the week old database.

Plan A

- The database .lg file showed the extents where the corruption was located.
- Each storage area was a single variable length extent
- Corruption was in an 80 GB extent (Ugh!)
  - Used DBRPR to scan and fix bad blocks
  - This took hours to run on this large extent
  - In the end this failed

Plan B

- Worked with the vendor to find all the tables that made up the report card processing
  - This was about 12 tables
- Dumped these tables from the live database
  - There was no corruption in these tables
- Had to figure out how to get the table data into the week old database

HMMMMM…..

Plan B

- Dumped the schema for the 12 tables
- Went into the dictionary and renamed the tables
  - Added _old to the end of the table name
- Loaded the schema for the 12 tables
- Loaded the data for the 12 tables

- This is a very useful trick
  - Didn't need to recompile – the application worked

Plan A (revisited)

- Dumped and Loaded the plan A database
- There were 5 tables where the dump and load failed.
- Did a 4GL dump
  - FOR EACH … BY field. EXPORT…
  - FOR EACH … BY field DESCENDING. EXPORT …
- Didn’t trust the data, so we use the same table rename technique to get these tables from the week old backup.

APJ
PROGRESS SPARK

But Wait – There's More

- A week later they found they also had corruption in a different database
  - That was solved by restore and roll forward
  - Needed to upgrade to 10.2B08 for Roll Forward to work properly

Next Steps

- Implement a DR solution
  - OpenEdge Replication
  - Rolling Forward AI
- Restore the backup and roll forward on the same machine
  - This verifies the backup is functional
  - DB block corruption does not get replication from roll forward

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

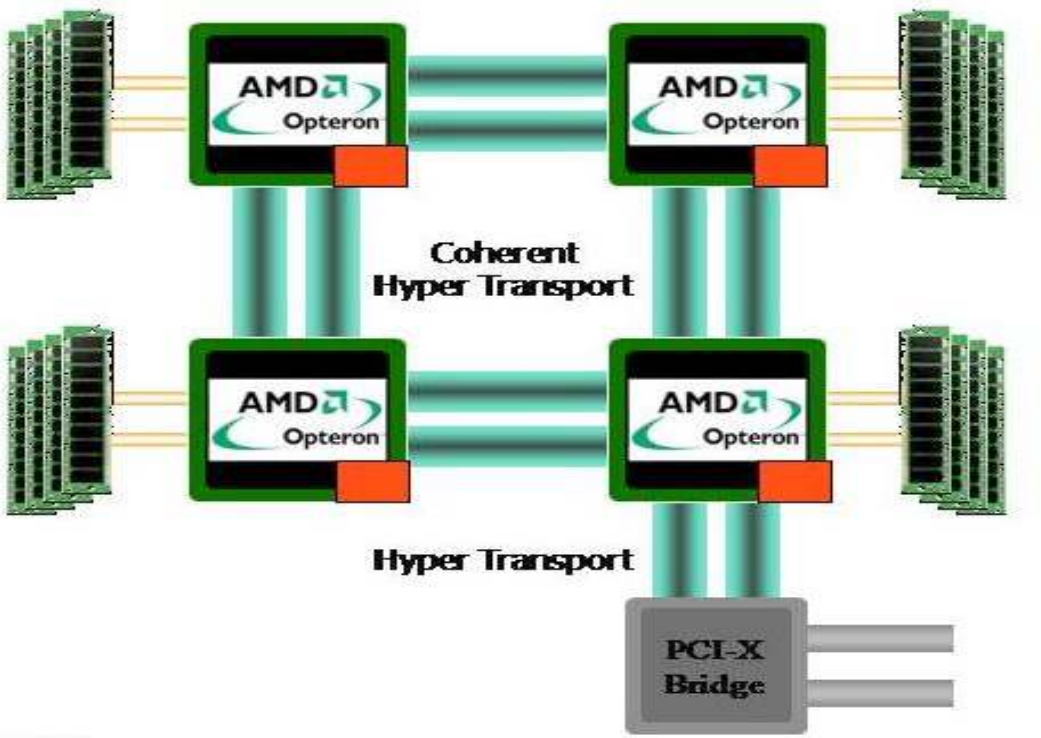
APJ
PROGRESS SPARK

Performance Problems

- Case Study 4
  - A customer recently purchased a new Linux machine to upgrade their old Linux machine. The new machine has more memory, faster disks, and twice the CPUs.
  - Their application runs slower on the new hardware.
- Specifics
  - CPUs are not busy
  - System Load is high (60 to 80)
  - Application is sluggish
  - Cannot identify a process causing the system load to go up

NUMA

- Non-Uniform Memory Architecture



APJ
PROGRESS SPARK

Cache Coherence

- OpenEdge needs to gain exclusive access to a region of shared memory
  - A test is performed to see if the region is locked.
  - This test requires all the CPUs to stop what they are doing, synchronize their cache-lines to make sure that multiple processes are not thinking that they obtained the lock at the same time.
  - As CPUs scale out wide, the Cache Coherency problem gets worse

Far Memory Reads

- Shared Memory Region spans nodes
  - Far Memory Access is commonly referred to as the NUMA Ratio
  - This is the time difference it takes to access memory from a CPU on node 1 to memory on node *n*. When the ratio is 1 it means you are on a SMP machine, when it's higher than 1, then you are on a NUMA machine.
  - Typical NUMA ratios are 3:1 or higher. This means that it takes 3x longer the access memory on a remote node than it does to access memory on a local node.
  - Given that database systems like OpenEdge, Oracle, MSSQL, etc access memory extensively, this 3x slowdown becomes a noticeable bottleneck.

Solutions

- Change operating mode to client/server
  - Less processes directly connected to shared memory
- Disable CPUs
  - Helps Cache Coherence problem
- Purchased a new machine

Agenda

- Disasters Area
- Performance Problems
- Migrations and Upgrades

YOU GET TO PIC THE NEXT MIGRATION PICTURE

PROGRESS SPARK

Migrations and Upgrades

- Case Study 5
  - 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

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

APJ
PROGRESS SPARK

Application Upgrades

- Case Study 6
  - 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 is a pain.

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 real world 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


APJ
PROGRESS SPARK