

XTIVIA

Continuance of Operation for Oracle Database Systems

Common Risk Scenarios, Mitigating
Technologies, and General Guidelines

by Andrew Dalby

V 2.0

May 2008

Introduction

CONTENTS

2	INTRODUCTION
3	RISK SCENARIOS
13	TECHNOLOGIES
21	GENERAL GUIDELINES

There are many types of service interruptions that can strike a business enterprise. To the extent possible, the ones that can be foreseen should be anticipated and dealt with proactively. A Continuity of Operations Plan (**COOP**) identifies risks and attempts to lay out a procedure to mitigate them. Of course, disasters by nature do not go according to plan, so a successful COOP will have layers of redundancy and allow for flexibility in implementation.

Likewise, there are risks that can be identified, but that are too costly to mitigate, or deemed so unlikely to occur, that the appropriate action is to do nothing. Thus, a COOP may cover scenarios spanning the gamut from “the DBA goes home for the night” to “a killer asteroid destroys the planet Earth.”

Critical business systems have been computerized long enough to draw some basic conclusions as to the frequency and impact of many scenarios. While each business must weigh its own circumstances and budget in determining which contingencies to cover, this document attempts to lay a basic groundwork spanning a broad swath of potential service interruptions, along with the technological solutions available for an Oracle database and some general guidelines.

Risk Scenarios

This section discusses many of the potential risks – some of which should be expected to occur frequently, while others may occur less frequently or possibly not at all barring some other catastrophic event.

Routine Events

People have an uncanny ability to focus on the rare and spectacular, while completely missing out on the mundane and common. As a result, many Disaster Recovery Plans focus on terrorist attacks or malicious hackers and completely ignore the most common causes of service outage. There is a class of outage events that can be categorized as business as usual, and every business should expect these events to occur frequently and plan appropriately.

Human Resources

The most mundane, yet high risk and costly, of all potential sources of service outage are simple personnel issues. DBAs are among the most highly compensated IT employees. It is not unusual for a skilled DBA to make six figures. Factor in benefits, taxes, and overhead, and the total cost of a DBA can be a quarter of a million dollars per year. Add to that, the fact that DBAs often don't appear to be doing anything since the job entails maintaining the status quo, and many organizations are hesitant to hire enough DBAs to ensure full-time coverage. There are 168 hours in a week. Assuming a full-time employee works 42 hours a week, never has a day off, gets sick, takes vacation, requires training, or quits, it requires 4 DBAs to ensure 24x7 coverage. Some enterprises are large enough that a team of 6 or more Oracle DBAs is a reasonable investment.

- ➔ *For those that do not have the budget to support this staff, it may make sense to invest in an outsourced DBA contract – at least to cover for the in-house DBAs when they are unavailable.*

Space

The nature of most databases is to continually grow. Certainly, the log files that are by definition always expanding, can easily consume all available disk space quickly.

- ➔ *Some plan must be made to monitor space consumption, and offload or delete log files periodically, as well as add disk if it becomes needed.*

Performance

Along with ever increasing space requirements, databases by nature add features and data, both of which tend to degrade performance. However, users have been

“Assuming a full-time employee works 42 hours a week, never has a day off, gets sick, takes vacation, requires training, or quits, it requires 4 DBAs to ensure 24x7 coverage.”

conditioned to expect performance to increase over time. There is a risk that an architecture will lock the system into poor performance or prevent upgrades to newer and faster hardware.

- ➔ *Being aware of the limitations of an architecture can ensure timely upgrades and performance.*

Maintenance

Microsoft and Oracle both release patchsets every quarter as well as emergency patches as needed. All non-trivial systems have bugs that are discovered and fixed. Hardware and software become obsolete and are de-supported. A database system will regularly need maintenance that will require the system to be rebooted. How to handle this is dependent on the organization. For those organizations that cannot tolerate downtime, clustering technologies can come into play. (These will be discussed in the technology section of this paper.)

- ➔ *A regular scheduled downtime is often sufficient. Occasionally, an emergency outage will be required.*

Testing

It is axiomatic that all changes should be tested prior to deployment in production. The risk of deployment without testing is that a minor change can have unanticipated cascading effects that can bring down the entire production system, destroy performance, or corrupt vital business data.

“[S]ome method must exist both for propagating changes [] into production as well as refreshing the test environment when it is stale or gets destroyed by an unsuccessful test.”

The Robust Testing Plan

A robust testing plan would have an identical setup to production on which every change would be made with a full-scale test battery run against it prior to a commitment to deploy. Unfortunately, this again doubles the cost of hardware and software without any performance gain which is frequently an untenable proposition. Again, the organization must weigh the risks versus the costs in determining whether to implement a testing platform, how to scale it, and how to test against it.

The Common Solution

The common solution is to use the old hardware for a test platform, while upgrading the production environment to new, more powerful hardware. While this is better than having no test environment, scalability, performance, and platform-specific issues cannot be accurately tested.

Failover Site as a Possibility

Another possibility that can reduce the overall cost is to have a failover site that is also used for testing. Since the failover site must be sized large enough to accommodate the production system, a full-scale test can be run in the environment to give good predictive results. The disadvantages to this setup are that in a high-transaction environment, it may be difficult to keep the standby

system up to date while also performing testing and that if a site failover occurs; all testing must be stopped, potentially delaying important features and idling a large development team.

Conveniently, the production backup can be restored to the test environment periodically. This validates that the backups are restorable, gives the team practice at the restoration, and keeps the test environment relatively up to date.

➔ *In any case, some method must exist both for propagating changes forward into production as well as refreshing the test environment when it is stale or gets destroyed by an unsuccessful test.*

Common External Failure

The next class of events that can cause service outages are also frequently overlooked, since they do not involve the loss of a computer component. They are however, not a normal part of business, so they are properly classified as failures, but are common enough to be expected to occur at least occasionally.

“... common enough to be expected to occur at least occasionally.”

Human Error

The most common source of data corruption is not hardware or software related. It is human error. This could be in the form of an operator mistyping data entry, a programmer's bug, or a database power user forgetting a WHERE clause in an UPDATE or DELETE statement. In all these cases, most of the database is unaffected, while one or two tables may need to be restored to a point preceding the mistake. Usually, shutting down the database while a full restore from backup takes place is unacceptable, but for many systems, that is the only solution to this type of problem.

➔ *Oracle has two additional technologies that allow for a speedy and low impact recovery from accidental data corruption – (1) the export / import / datapump utilities and (2) the flashback query, discussed in the technologies section.*

Network Outage

Network outages are an all too common occurrence. Managing a network of potentially thousands of computers is an incredibly complex undertaking, with many single points of failure. Even with a well managed infrastructure and appropriate levels of redundancy, networks out of the control of the enterprise, such as the ISP or even major backbone providers may go down.

- The internet worm of 1986 essentially took down the entire internet, and more recent attacks have substantially degraded its performance on a world-wide scale.
- Farmers, construction crews, and homeowners dig up buried cables; animals, trees, and storms knock out above ground ones; hackers, communications companies, and governments disrupt traffic on purpose or by accident.

Planning for network outages is very difficult, if for no other reason than our ability to determine the scope and nature of a problem relies on the very network that is down. In true national disasters like hurricane Katrina, it can take days or weeks to fully comprehend the extent of the damage.

Unfortunately, there is a limit to the technological fixes that can be applied to network outages. A distributed architecture can offer some mitigation in some circumstances, but it is easy to end up with a “split-brain” if two distributed nodes can’t talk to each other and both assume they should be active and in control. The marketing of replication is seductive, with the promise of geographically dispersed, “location-free computing”, but in reality, replication is very difficult to set up, almost impossible to keep up for any period of time, and puts an enormous burden on the slowest component of the system (the network). There are situations where replication is the correct answer, but they are very rare. Nonetheless, replication technologies will be covered in more detail in a later section.

- ➔ *Ensuring adequate redundancy is a good first step, and may be the only appropriate technological one.*
- ➔ *For any unanticipated problem, the correct choice may be to rely on the network administrators to diagnose and fix it on the fly.*

Power Problems

In contrast to networks, the power grid in the United States is arguably the most reliable major system in existence. Generators are synchronized with atomic clocks, so the phases in the AC supply are used as exceedingly accurate timing pulses in a variety of applications. Common experience leads us all to expect the lights to turn on every time we flip the switch. However, even 99.99% uptime means a system is down for nearly an hour every year, and the vaunted six-nines (99.9999%), considered by many to be the highest attainable reliability means the system will be down for an average of 30 seconds every year. Add in power surges and fluctuations, and it is imperative to put some manner of power protection in place.

“99.9999% considered by many to be the highest attainable reliability means the system will be down for an average of 30 seconds every year.”

- At the very minimum, a \$20 surge suppressor power strip is required.
- More realistically, a good UPS is called for. It must be rated to support the wattage of the server for an appropriate length of time (probably an hour or so), and be able to initiate an automated graceful shutdown.

However, just purchasing the UPS is not a magic bullet—the batteries must be changed regularly, and it must be tested including the shutdown routine. It is not uncommon for a vital system to crash due to an improperly installed and/or managed UPS.

To ensure power protection beyond the lifetime of the UPS batteries requires a secondary power source, which could be a secondary grid hookup from the utility company or a diesel generator. If a diesel generator is chosen, it too must be tested and

maintained correctly, must allow for refueling during operation, and sufficient diesel fuel must be maintained or available, which bring along the problems associated with storing hazardous, flammable materials.

Even if good power policies are maintained, simple mistakes can result in power outages. Most IT veterans can tell a story about someone who accidentally unplugged a production server. The only effective countermeasure for this type of outage is methodical work and effective management.

The vast majority of the time, if the power to the database server dropped unexpectedly, every time Oracle starts a database, it will simply fix any inconsistencies and start fine, though a bit slower due to the time spent on consistency checks. Occasionally, a file may be corrupted because the disk told Oracle a write was completed when it wasn't. This is typically caused by write caching, which should always be turned off on any database system. The only exception is on high-end SANs that have highly redundant battery-backed up cache. If a data file is corrupted, a full restore and recovery is needed, which will be covered in a later section.

➔ *Fortunately, even if the power to the database server drops unexpectedly, Oracle provides very good automatic crash recovery that is run every time Oracle starts a database.*

HVAC Troubles

A typical pocket calculator can run off the electricity generated by a 60W bulb shining on a photovoltaic cell less than one square inch. Indeed, 20 years ago, the power consumption of a desktop computer was fairly trivial. However, as Moore's law has caused computers to become more powerful and processors to shrink, it has caused a corresponding increase in power consumption. In a server room, a half-dozen racks of 2U servers could be the equivalent of a 50,000W electric heater. This means that if the HVAC system goes out, the systems may start to overheat and shut down in as little as 15 minutes. Most of the problems associated with HVAC shutdown are similar to an unexpected power outage, and in fact, power outages are frequently the cause of HVAC problems, since the UPS and generator are rarely sized to include the environmental power needs of the server room.

➔ *The UPS and generator used to serve the systems in the server room should be large enough to support the HVAC system during a power outage.*

Security Breaches

We have finally gotten to the class of events that are commonly recognized as threats, though they still are not a failure of technology. It is important to remember that a security breach cannot be undone, so while litigation may recover damages, mitigation is a matter of prevention.

“It is important to remember that a security breach cannot be undone ... mitigation is a matter of prevention.”

Physical Loss of data

When we speak of data loss in technology, usually we mean that the magnetic media on which it was written somehow is unable to be read. However, the media itself can be lost or stolen. Especially of concern in this area are backup tapes, laptop computers, and thumb or external drives. There are many legitimate reasons a user may transfer data to a portable device, but once the data leaves the data center, it is impossible to keep secure. A policy should be crafted that explains when, where, why, and how portable devices are to be used and it should be strictly enforced. Any perceived need to circumvent the policy should be dealt with, or employees will simply ignore the policy.

Perhaps more worrisome are backup tapes that are usually created in the data center for the express purpose of moving off-site. Frequently, these tapes contain the most critical enterprise data in unencrypted format. Their loss or theft could jeopardize the entire organization and have legal ramifications as well.

- ➔ *All data backed up to tape should be encrypted if it is to be transported out of the data center.*
- ➔ *Alternatively, direct backup across a secure network to disk at a remote site can eliminate this risk.*

Rogue Employee

The vast majority of security breaches are not by an external hacker breaking in, but are committed by employees. They may be motivated by greed, revenge, or just trying to “get the job done.” First and foremost, you must be able to trust your employees, particularly your DBAs and System Administrators (SAs).

- DBAs have unlimited access to everything in an Oracle database. They can view everything, change everything, and make it look like someone else did it.
- SAs control the domain accounts and file systems hosting the database. They can alter users, add themselves to the DBA group, and surreptitiously copy data or log files, install sniffer software, or key loggers.
- Either can simply shut down the database or destroy data files.

Therefore, at the very least all SAs and DBAs should be subjected to a thorough background check.

Auditing

Auditing should also be turned on, and should be put under control of a different team. This segregation of duties may be required for compliance with certain regulations such as SOX, PCI, and HIPAA. Normal Oracle auditing takes place within the database, which of course is inadequate for monitoring the DBAs, but more convenient for reporting. OS auditing can keep the audit trail out of the control of the DBA, but puts it within reach of the SA and the DBA can still turn auditing off. A third-party network-based auditing appliance such as Guardium

SQLGuard may be the best solution to this need. While this type of auditing system is somewhat expensive, it may be justified. Of course, while auditing can discourage unauthorized actions, it cannot stop them.

Proper Controls

Proper access controls and the principle of least privilege are important. Oracle can enforce a password complexity requirement, and multiple profiles can be set up, with independent rules. It is important to realize that the most powerful users often have the worst passwords. All the default passwords should be changed, and where possible, system accounts should be disabled and locked.

Administrators should not share accounts, and should be required to follow the password complexity requirements. No user should be granted any more permissions than is necessary for their job, and applications should allow them to do what is needed, so workarounds that violate security aren't seen as necessary. If an application allows a user to make mistakes that require manual intervention by the DBA, it should be redesigned, to prevent the mistake or allow correction within the application.

- ➔ *By instituting various checks and balances through audits and establishing proper access controls for all users, intentional and unintentional breaches by employees can be minimized.*

Malicious Hackers

The same guidelines apply to securing a database against hackers as against employees, but since hackers may be more adept at breaking in, some additional precautions are warranted. While it is common to design applications that enforce security and log on to Oracle as a common user with ownership of the application objects, it is discouraged. There is a saying in the security community that such an architecture is “like driving a big stake in the ground and hoping your adversary decides to drive into it.”

“... if a hacker chooses to bypass [an application's security], there is no security at all.”

No matter how well the application implements security, if a hacker chooses to bypass it, there is no security at all. To provide optimal security, schemas designated to hold objects should not allow logins. Instead, a low privilege account should be designated for logins, and data access should occur through stored procedures. Of course, this places business logic in the database, and requires application programmers to be able to write PL/SQL, which may not be the right choice for business reasons.

- ➔ *Regardless of architecture, SQL injection needs to be carefully guarded against, and all data access code should be reviewed by a DBA to ensure it cannot be used to elevate privileges.*

Hardware Failures

The last class of events that could cause a service outage are failures of part or all of the physical system.

Tape Failure

While tape is no longer the predominant medium for online data storage, it retains that position for backups. However, it is losing ground in that area as well. Historically, the serial nature, slow access times, and manual intervention requirement of tape were seen as less important than its low price and high capacity. However, with shorter maintenance windows and the price of other media dropping, these advantages are no longer as compelling as they once were. Indeed, the fact that tape is far more likely to fail than disk (Gartner research has reported that up to 1 in 10 tapes will fail) make blind reliance on tape for backups unwise.

➔ *A now common technique is Disk to Disk to Tape (D2D2T), where the initial backup is made to disk, and that backup is copied to tape – this is an extremely attractive option that allows for quick recoverability and reliability, while keeping prices low.*

Drive Failure

Database systems typically are I/O bound, meaning that their disks are spinning most of the time. With such a load, high quality disks are a requirement. SCSI or SAS disks will outperform IDE or SATA drives that simply cannot handle the constant use. As mentioned in the *Power Outage* section, *disk caching should be turned off to prevent data corruption.*

RAID

RAID technologies are also important, but should be configured as RAID 1 (mirrored) or RAID 10 (striped and mirrored). Trying to achieve higher space efficiencies by using RAID 5 (striped with parity) imposes a parity calculation on every disk access, and loses the benefit of parallel reads.

SANs

SANs are a popular way to get redundancy and reliability and are recommended for those organizations that can afford them, though their prices have come down to the point that only the smallest shops can legitimately use that argument.

Of course, the DBA and SAN administrator should negotiate an adequate solution for disk speed and redundancy. LUNs used for database applications should not be built on disks shared with other applications, since that can adversely affect performance. Oracle itself will perform mirroring of certain files, but except for the control file, it is generally best left to hardware. Files, which are irretrievably lost due to multiple disk failures, will require a restore and recovery process, and most likely will experience some data loss.

Alternatively, the server can be written off as a loss and rebuilt, while the business relies on Server Failure strategies and technology for continuance.

- ➔ *Working together, the DBA and SAN or System Administrator should determine the optimal hardware solution to protect critical information from being lost due to disk failure.*

Solid State Failure

The failure of any solid state component (CPU, RAM, NIC, Motherboard, etc.), is typically dealt with at the level of the BIOS, which should either compensate by switching to a redundant component or shut down the server. As such, no particular preventive measures are called for with regard to Oracle.

- ➔ *Purchasing high-quality servers with redundant components and self monitoring is the best protector against this risk.*

Server Failure

If enough of the redundant components in a system fail or in the case of spill, fire or fall, an entire server can fail while other systems at the site remain operable. The strategies and technologies for dealing with a Server Failure are mostly the same as those for *Maintenance and Testing*. Depending on budget as well as downtime and loss tolerance, the choice can be to wait for a replacement system (days), restore to a pre-built and dedicated backup (hours), failover to a standby (minutes), or failover to cluster (seconds).

- Ordering a new server with express shipping can typically have replacement hardware ready for use within 2 days. Spare parts from a hardware vendor will take several hours to be delivered and installed, even with premier support. If maintained on-hand, spare parts can be installed more quickly, but not likely in less than an hour or two. Reinstalling and patching the OS and DBMS will normally take at least a day.
- Restore and recovery operations can take many hours, especially if backups are stored off site and need to be retrieved.
- Failover to Standby is usually a manual process to prevent accidental failover, so the limiting factor is typically the time it takes an administrator to login to the box and initiate it.
- Failover to cluster is automatic and takes seconds or no time depending on the cluster setup.

Data loss follows a similar pattern.

- Daily exports or backups mean the exposure to a full day's loss.
- Intra-day archive log backups includes the potential of loss for the duration between log backups plus the amount of data that is in the online redo logs not yet archived.
- Oracle-managed, archive log duplexing eliminates the scheduled exposure, but does not provide coverage of the online redo logs.

- DataGuard set up in LGWR mode eliminates even this potential for loss. If set up asynchronously, only the data in transit is at risk. If set up synchronously, DataGuard prevents all data loss—but at the cost of availability.

To ensure zero data loss, synchronous DataGuard requires all logs to be written to both the primary and secondary site before allowing the transaction to be completed. If either system is unavailable, or if the network connection goes down, all transactions are hung until the problems are resolved.

➔ *FailSafe clustering provides zero data loss and seconds of downtime due to shared disk, while RAC clustering provides zero data loss and zero downtime due to its grid nature.*

Site Failure

The most comprehensive failure usually covered in a COOP is Site Failure. Not only is such an event extremely rare, since it is typically caused by an act of war or God, it is also very expensive to mitigate against, since all infrastructure must be duplicated.

Also, due to the probable scope of impact, personnel losses should be anticipated when planning for a Site Failure. It is quite likely that an event that causes the loss of a facility will also include the death, or disabling of key individuals. Even if none of the key employees are directly affected, they may have moral or legal obligations to care for those who have been, and will therefore be unable to act in their normal capacities much less the extended capacities needed for disaster recovery.

Thus, when planning for the eventuality of site failure, the redundant site must also include people to run it. Even with full site duplication, it is possible for multiple sites to be lost in close succession, so it does not provide absolute assurance. Many companies who had redundant data centers in New Orleans and Houston found this out during the hurricane season of 2005 when both cities were hit by major hurricanes (Katrina and Rita) within weeks of each other. As mentioned in the introduction to this paper, in the extreme case of a killer asteroid destroying the planet Earth, all possible redundant sites would be destroyed, and so no mitigation strategy is even feasible.

“When planning for the eventuality of site failure, the redundant site must also include people to run it.”

Technologies

This section discusses various technological components within Oracle that are available to manage risks.

Security

As mentioned in the *Risk Scenarios Section*, HR issues are a more likely cause of corruption or breach than technology ones. The use of security techniques is critical to reduce these risks.

Privileges

The most basic security technique is also the most critical. Don't let people do what they shouldn't. This idea, called the Principle of Least Privilege, simply means give only the permissions needed for the job at hand.

Oracle has a very fine-grained and sophisticated permissions model. It should be used. Do not grant DBA to anyone who isn't one. Do not log on as the schema owner for an application. Create individual user accounts and grant specific permissions to each user based on what the user needs to do.

"... only give the permissions needed for the job at hand."

➔ *Fully utilize Oracle's permissions model to integrate the Principle of Least Privilege with all users.*

Policies

Oracle Policies are configurable limits on what users can do. They can enforce password complexity and change requirements as well as limit resource usage.

➔ *Like Privileges, appropriate use of Policies will keep your Oracle database secure.*

Auditing

Users will frequently ask the DBA who changed a particular data value. Unfortunately, by default, Oracle does not keep track of this information. Auditing must be set up and enabled to track who did what. In addition to being a good idea from a security standpoint, auditing may be required for regulatory compliance. SOX, PCI, and HIPAA are the big regulations in the United States requiring auditing.

Basically any firm that keeps medical or credit card data or is publicly traded is legally obligated to have auditing in place. Oracle does have auditing abilities. But due to its architecture, Oracle is limited in protecting against a malicious DBA.

- ➔ *For better protection against a malicious attack, it is probably better to use a dedicated auditing appliance such as Guardium SQLGuard.*

Logical Backup

The most basic COOP technology that requires practically no resources and that are almost impossible to implement incorrectly are logical backups. Logical Backups provide some level of protection against almost all of the failures listed. A daily export of the database requires only some spare disk space, requires no downtime, and is transparent to the users of the database, including other backup strategies. The files are easy to transport to alternative storage that can be geographically dispersed. Virtually all production environments should have a logical backup in place.

“Virtually all production environments should have a logical backup in place.”

Export / Import

Oracle’s export and import utilities have been around for a very long time, are well tested, and extremely flexible. Export creates a dump file that is essentially just the series of SQL statements needed to recreate the database as of the point in time it was run. As such, the dump file can be used to migrate from one version of Oracle to another; or from one platform to another. However, because the dump file is essentially a text file with no indexes or pointers, it can take a very long time to parse just to find the table affected.

Once the data is found in the file, the SQL statements are run through the standard SQL engine, which parses them, generates undo and redo logs, allocates memory buffers and performs security and integrity checks, all of which also introduce overhead. It can be run against the database as a whole, certain defined schemas, or just certain tables. It can export the structures only, or include the data, indexes and constraints. It is also platform and version independent. However, since it is a snapshot, it keeps no historical record and cannot capture incremental changes.

Import is the consumer of a dump file and is similarly flexible in what it can import. These tools are particularly useful for duplicating a table, schema, or database for testing or migration and for recovering from human error.

- ➔ *The significant disadvantages of a dump file are that it is non-transactional, the utilities are slow, and if a job fails it must be rerun in its entirety.*

DataPump

DataPump is a newer utility that Oracle released to address some of the limitations of Export / Import. DataPump outputs to a binary file that is fast searchable and can import directly to the data files bypassing the SQL engine and associated overhead. DataPump can also interrupt and resume a job.

- ➔ *Unfortunately, DataPump still creates a non-transactional logical backup, which means that like Import, it can bring back all the data that existed as of the point in time when it was run, but any data entered since then is lost.*

Flashback Query

Starting in version 9i, Oracle introduced a feature called Flashback Query, which allows the database to be queried as of a previous point in time. While not technically a logical backup, it protects against many of the same risks and can be viewed as a form of continuous logical backup. In a typical scenario where a user forgets a WHERE clause and accidentally deletes an entire table, the Flashback Query can be used to simply INSERT into the affected table the records SELECTed as of five minutes ago.

- ➔ *Flashback Query needs to be activated and configured properly, and it depends on logs remaining on the server long enough to cover the flashback period, but is ideal for protection against the typical database “oops”.*

Physical Backups

An Oracle database consists of several types of files, all of which need to be backed up and restorable to provide protection against failures. There is a common misconception among system and backup administrators that all servers are alike and that the backup strategies that work for file or web servers are applicable to database servers. This couldn't be further from the truth.

“An Oracle database . . . [must] be backed up and restorable to provide protection against failures.”

File and Web Servers

File and web servers tend to have a large number of small files that are rarely updated. An incremental backup will catch the few changes and save a large amount of space over a full backup. File handles can be relied on for protection, since file changes are rare, quick, and independent of each other.

- Simply copying files to backup media, waiting for file locks to clear is adequate for these servers.

Database Servers

Database servers are entirely different. By their nature, database servers have a few very large files that are constantly open and being written to in many places at the same time.

- Since the only application that should ever be allowed to touch a data file is Oracle, and since Oracle has a multithreaded / multiprocess architecture, it does not acquire blocking file locks.

- Likewise, since data files are usually multiple gigabytes in size, changed frequently, and in coordination with each other, a naïve backup strategy simply will not work.

Given these differences, (1) incremental backups do not save space since every file has changed, (2) reliance on file locks fails since Oracle doesn't bother with them, (3) file contention between the backup software and Oracle can take place, and (4) the sheer size of the files ensures that hundreds or thousands of file updates will take place during the time it takes to copy a single file.

Unfortunately, since typical backup tools don't recognize these limitations, simple backups will appear to work until they are needed and an attempt is made to restore them. These "backups" are nothing more than wasted tape, since they are completely unrecoverable. For an exorbitant fee, Oracle Professional Services will process them and can potentially retrieve a portion of the data, but the odds of a full or even majority recovery are essentially nil.

➔ *Physical backups are the foundation of any sound backup and recovery strategy.*

Archive Log Mode

The greatest advantage Physical Backups have over Logical Backups is the Point-In-Time Recovery. There is an old joke about legislators trying to ensure fire safety. They are concerned that fire extinguishers may lose effectiveness over time, and so they require that they are tested 15 minutes prior to any fire. DBAs face the same dilemma the lawmakers did. We can protect against disk failure by copying the data to a different device, but as soon as the copying is done, it is obsolete. Ideally, we would want to take a backup immediately prior to a disk failure. Since we can't predict failure, Oracle does the next best thing and logs everything before it is actually done. As long as we have an unbroken chain of logs, we can reconstruct a database without losing anything.

The Archiver is the Oracle process responsible for copying current logs to the archive, which should be done on all production databases. Of course, replaying every log file from the creation of a database would be unreasonably slow and tedious, so we perform physical backups as a baseline from which the logs can be applied.

➔ *Physical backups allow for logs that predate the backup to be discarded while allowing for reconstruction from time of physical backup until the current state of the database at the time of a crash.*

Cold Backups

To correctly back up a database server, the files need to be quiesced, preventing further writes while the file is copied. There are three different ways to accomplish this. Cold Backups are the simplest and most foolproof way to perform a physical backup. A Cold Backup simply shuts down the database, allowing the instance to complete all active

work and exit. Once the instance has stopped, all files are copied using OS utilities. The disadvantage of the Cold Backup is of course, that the instance has to be down for the length of time it takes to copy the files. In the days of small databases that could be shut down in the evenings, this was not a large sacrifice. But may not be practical today.

➔ *One way to perform a physical backup, Cold Backups are best for small databases with regular downtimes.*

Hot Backups

As more and more companies strive to operate 24x7, a large daily maintenance window may be infeasible. The solution to this problem is the Hot Backup, which allows the files to be copied without shutting down the database instance. Of course, we still need to stop all writes to the file we are copying. In this case, Oracle is configured to place a tablespace in backup mode, deferring writes to the file until the tablespace is returned to normal mode. Since all transactions are logged before they can be committed, it is possible for Oracle to postpone writes for a while, and catch up from the log once the file copy is complete.

Naturally, the additional complexity introduces risks. Since tablespaces are typically taken offline one at a time, the data files do not form a consistent snapshot and the logs written during the Hot Backup are critically important. A database must be running in Archive Log Mode to place a tablespace in backup mode.

➔ *When using Hot Backups, care must be taken to ensure that the files must be placed back into normal mode as quickly as possible.*

RMAN

With both Cold and Hot Backups, data files are copied in their entirety, which may not be the most efficient strategy. Frequently, the bulk of the data is unchanged, and copying only the changes is preferable. As discussed previously, OS level tools cannot perform this differential backup since they are unaware of the nature of the changes in the data files. Oracle has provided a utility that can backup data on a block-by-block basis rather than on files as a whole. RMAN can also encrypt and compress backups, create backup files of arbitrary size and run in parallel. It is highly recommended that RMAN be used whenever a Physical Backup is called for.

➔ *RMAN is Oracle's Recovery Manager, a tool that integrates with sessions running on the Oracle server to perform a range of backup and recovery activities, as well as maintaining a repository of historical data about your backups.*

"Oracle has provided a utility that can backup data on a block-by-block basis."

Replication

Replication is similar to backups in that a copy of the data is put in a different location. It differs in that the data is put into a database that can be used quickly if there is a

problem. In the best-case scenario, restoration from backup will take at least a day. Assuming a spare system is available, it takes around 4 hours to install the OS and about the same for the DBMS. Creating a new blank database takes about an hour, while physically copying the backup and restoring the data can take from 2 hours to several days depending on volume. If the tape needs to be recalled and manually inserted into the silo, another day is expected. If the hardware needs to be requisitioned, days or weeks will be added. Conversely, for replication, a duplicate system is pre-purchased; OS and DBMS are pre-installed and configured to mirror the production box as closely as possible. Oracle provides a number of methods to move data to the backup system depending on need.

DataGuard

The simplest replication model is log shipping, where in essence the replication is merely the backup and logs that would be committed to tape instead being transferred to a prepared backup system. Since the system is prepared and the backup files are kept locally on disk, only the restoration and log application needs to be performed, dropping the outage time to perhaps an hour.

DataGuard takes that a step further by pre-restoring the database and continuously applying logs as they are generated on the primary system. This further reduces downtime exposure to perhaps 5 minutes. In fact, in a DataGuard system, the majority of downtime experienced in a disaster is the time it takes a DBA to verify that the primary is down and log into the standby instance.

➔ *Data Guard maintains the standby databases as synchronized copies of the production database.*

Logical Standby

An alternative configuration of DataGuard, Logical Standby translates the logs to SQL statements and issues them to an open duplicate database. This allows the standby to not only cover for the primary in case of failure, but to serve as a reporting instance offloading some of the burden from the primary instance for read-intensive operations. Logical Standby can also filter out certain statements or add business logic to manipulate data as it is moved, which can be beneficial to a reporting server.

➔ *The caveat with Logical Standby is that the instance is open for read/write operations and it cannot be guaranteed that no one has manipulated the data on standby in a manner inconsistent with production.*

Materialized Views

Another replication technology that can behave in a similar manner to a Logical Standby is Materialized View Replication. A Materialized View is created in the replicated instance that pulls data from the primary system.

- ➔ *The primary difference between Logical Standby and Materialized View Replication is push vs. pull.*

Multimaster

To allow for bi-directional replication, Oracle has Multimaster Replication. Significantly more difficult to set up and administer than either DataGuard or Materialized View Replication, Multimaster Replication also requires significantly greater infrastructure. However, in exchange for this added expense and difficulty, Multimaster Replication allows for Geobalanced applications, where multiple sites exist not only for disaster protection, but also for load balancing.

- ➔ *Due to the complexity of Multimaster Replication, it is not recommended except in very unusual situations.*

Streams

Adding another level of complexity, Streams Replication uses Queuing and dispatching to keep multiple heterogeneous databases apprised of specific changes. For example, an HR system and a GL system may want to be aware of certain changes in each other, but not care about others. Streams can ensure that they are kept appropriately synchronized.

- ➔ *It is possible to use Streams Replication for COOP purposes, but due to its complexity, this is not recommended.*

Clustering

Taking Replication a step further, so that no manual intervention is required at all for failover, Clustering turns multiple systems into a single virtual one, potentially reducing failure downtime to zero. Of course, the intercommunication between nodes in a cluster are so time dependent that all nodes of a cluster need to be in the same room, so this technology does not provide protection from Site Failure. Oracle supports two types of clustering: FailSafe and Real Application Clusters (RAC).

“Clustering turns multiple systems into a single node . . . this technology does not provide protection from Site Failure.”

FailSafe

FailSafe is an extension on top of Microsoft Clustering Services (MSCS). Obviously, as such, it is only available for Windows systems. The failover mechanism is robust and automatic, but does require a few seconds to complete and any existing connections are dropped. Compared to other clustering technologies, FailSafe is relatively inexpensive and easy to set up and maintain, but it also suffers from many of the common negatives.

Rather than a single server, two identical servers must be acquired and licensed, doubling the hardware and software costs without adding any additional usable computing power. Setup and maintenance of a FailSafe cluster is more difficult, time consuming and expensive than for a single server instance, and data storage must be on

a SAN that allows shared control. However, it does allow one server at a time to be brought down for maintenance without requiring the database to be unavailable for the entire time. It also mitigates against server failure, which will be addressed later.

➔ *FailSafe is dependent on MSCS architecture, notably a restriction to two nodes, one active and one passive.*

RAC

RAC is Oracle's premier clustering technology. It is platform independent and provides for both disaster mitigation and load balancing. As many as 100 nodes can participate in a RAC cluster and the load is automatically balanced between them, with no interruption in service at all if a node goes down.

- Each node must be purchased and licensed, so a RAC can get quite expensive very quickly.
- Likewise, RAC is notoriously difficult to set up and maintain, so it is not a solution to be chosen lightly.

For instance, a four node Dell 2950 dual processor dual core cluster running Red Hat Enterprise Linux (RHEL) could easily compete on price and performance with an eight processor HP 9000, while also providing the advantages of clustering and the ability to grow the cluster with the addition of more commodity servers.

➔ *RAC is the only option that allows for zero downtime for maintenance or server failure, and may actually be cheaper than a large SMP system.*

General Guidelines

This section highlights the crucial steps of an Oracle database installation, whether you have a COOP in place or not.

Universal Recommendations

Without contradicting the common wisdom that all blanket statements are false, there are some guidelines that should be followed by every Oracle installation. Only if there is a clear and compelling argument to the contrary and it is impossible to work around those issues should these guidelines be ignored. They are also easy and cheap enough that anyone even contemplating using Oracle can manage and afford them.

“Anyone even contemplating using Oracle can manage and afford [these universal guidelines].”

➔ Security

Security should not be an afterthought. Applications must be designed with security in mind and development environments should be as secure as their production counterparts. Using appropriate permissions and policies are critically important and may be legally required.

➔ Daily Logical Backup

A Daily Logical Backup provides a baseline of security against disk failure, operator accident, and database migration. The export is performed online and is invisible to the users. The only price to pay is some disk space to hold the dump file. At least one dump file should be retained locally and should also be copied somewhere else—whether that be to tape or disk on another device.

➔ Archive Log Mode

Archiving the log files is the only way to recover a database to the point of failure and is required for Hot or RMAN Backups. Unless the work performed in the database is of no value, Archive Log Mode should be on.

➔ RMAN Backups

While the Logical Backup provides a baseline of security, a Physical Backup is needed to use the Archived Logs. Cold and Hot backups are tolerable, but RMAN has so many advantages over them that there is no reason not to use it.

➔ Virtual DBA

Oracle is one of the most complex pieces of software ever built; the documentation for the current release runs around 50,000 pages. Mere competence at Oracle takes years of experience that cannot be gained by a programmer or system admin doing DBA tasks on the side. Combine the difficulty of administering Oracle with the fact that it typically contains the most

valuable data in a company, and it is imperative that a team of experts be retained to manage it. While large enterprises may elect to do this work in-house, it is more cost-effective to outsource this function to a firm that specializes in DBA services, such as Xtivia. VDBA services can be had for as little as \$500/month, so even small shops can afford this vital service.

Basic Protection

This class of protection is an extension to the Universal Recommendations, and covers the most common risks within a reasonable budget for even smaller companies. If your company has progressed beyond the startup stage (meaning your CEO takes a salary and you are reasonably certain your paycheck will clear), these recommendations should be followed.

“If your company has progressed beyond the startup stage, these recommendations should be followed.”

➔ Flashback Query

Flashback Query requires a minimal set up and is slightly complicated, so it may be inappropriate for the smallest shops or development environments. On the other hand, it is so useful for recovering from operator error and debugging applications, it should be set up wherever possible.

➔ DataGuard

While a backup may keep your company from bankruptcy and you out of jail, the day to weeks of downtime required to restore a critical database is usually unacceptable.

- The cost of an idle day for an entire production floor or lost revenue and goodwill from being offline are usually orders of magnitude greater than the cost of a duplicate system with DataGuard running.
- Additionally, if implemented at a secondary site, DataGuard protects against virtually all foreseeable technology failures.
- Of course, DataGuard does require a knowledgeable DBA to manage it, so do not implement this feature without one on staff or on call.

➔ Development Environment

Developing or testing in production is a common but foolish risk that makes no more sense than simply handing the car keys to a small child and asking them to drive in heavy traffic.

- One strategy is to retain obsolete hardware as the development/test environment when upgrading production.
- A better way is to buy hardware in pairs and set up the development environment on the DataGuard backup server. Since the servers are twins, much more reliable test information will be obtained, and it will be able to support the application if primary goes down.

➔ Database Monitoring

If a database is critical to a revenue stream as most are, it should be proactively monitored. It is possible to create custom monitoring scripts, or perform this by hand, but it is much more cost effective to upgrade the VDBA service to include monitoring, if it isn't already included.

Midlevel Options

➔ Auditing

Auditing has been compared to closing the door after the horse is gone, but it has a deterrent effect and may be required for certain applications or organizations. It is a bit of a two-edged sword however, in that generating audit logs, but not reviewing them, can be viewed as negligence. Auditing should be turned on if required and may be turned on for business intelligence reasons. If done, a security officer should be budgeted at a minimum of 20 hours per week.

- An auditing appliance such as Guardium SQLGuard significantly reduces impact and effort at a reasonable cost.

➔ Reporting Instance

At high transaction volumes, it may be advantageous to offload the reporting/bulk processing to a separate instance to avoid impacting OLTP work. This should only be done after it has been shown that a reasonable investment in hardware will not allow reporting out of the production instance, due to the complexity and risks involved in advanced replication schemes.

- Designing an appropriate replication mechanism and ensuring that it is working properly requires advanced DBA skills and will require significant up front effort and between 10 to 80 hours per month.

Esoteric Extravagances

There are situations where the more esoteric features of Oracle are called for, but they are very rare and should be avoided when possible due to the expense and difficulty of implementation and maintenance. These features require months of design and planning and a minimum of 20 hours per week to manage. This section describes these technologies and the situations where they would actually be useful.

➔ RAC

While RAC is a spectacularly cool technology and grid computing is undoubtedly the future, virtually every Oracle DBA knows of at least one RAC failure.

Whereas a typical Oracle install takes around 4 hours, a RAC install takes at least a week. On the other hand, once a RAC is up, it is nearly bulletproof and

does provide a very high level of Load Balancing and High Availability. Situations where RAC makes sense are Million Dollar a Minute or Life and Death applications. A critical hospital system that simply cannot have downtime or people may die should implement RAC. A website that is a household name like Amazon, which needs to balance thousands of transactions a second and which a minute of downtime means major revenue loss, should implement RAC. For most other organizations, DataGuard and/or larger hardware are better solutions.

➔ **Multimaster Replication**

Multimaster Replication is the converse of RAC—it is fairly simple to set up, but can be counted on to fail regularly. The minimum infrastructure requirements for it to function reliably are in the \$100,000+ per month range and isn't even available in most markets. A dedicated Network Operations Center with redundant OC3 connections or its hosted equivalent is called for. As with RAC, Fortune 400 companies that are hosting extremely high volume, billion dollar applications may have a need for Multimaster Replication.

For More Information

To learn more about COOP as well as other Xtivia products and services, please visit www.Xtivia.com.



© Copyright 2008 Xtivia, Inc. All rights reserved.

Xtivia, Inc.
304 South 8th Street
Suite 201
Colorado Springs, CO 80905
(888) 685-3101, Option 2.