November 2004

DB2. Information Management Software

Why should you care about the cost of your High Availability solution?

IBM Software Group
Toronto Laboratory

Contents

Introduction

Application availability has always been a critical component of any enterprise-class, mission critical application. IBM continues to deliver high availability features to meet or exceed customer demand and DB2 UDB V8.2 is no exception. In fact, the new high availability features incorporated into DB2 UDB V8.2 can deliver higher levels of availability and at a lower cost than competitive products.

One of the cornerstones of the V8.2 release is the new High Availability Disaster Recovery (HADR) feature which is included in the base DB2 product for Enterprise Server Edition customers. Although this is the first release of this technology in DB2, it is technology that has been proven with Informix customers and is now available in DB2. Informix Dynamic Server (IDS) first shipped this high availability technology in 1993, and it has been deployed in numerous mission critical applications over the last decade. This proven technology is now available to DB2 customers and can deliver faster failover, with simpler management and lower costs when compared to other failover/clustering technologies available today. In fact, recent tests show DB2 with HADR is able to failover an SAP R/3 application running 600 simultaneous users to a standby server in as little as 11 seconds.

What is HADR and how does it work?

High Availability Disaster Recovery (HADR) is a data replication feature that provides a high availability solution for both partial and complete site failures. HADR protects against data loss by replicating data changes from a source database, called the primary, to a target database, called the standby.

A failure can be caused by a hardware, network, or software failure. With HADR, the standby database can take over the workload in a matter of seconds from any of these failures including disk failure on the primary. Furthermore, you can have clients automatically redirected to the standby database (now the primary database) without the need for changes to your application with the new automatic client reroute facility of DB2 v8.2.

Data changes made on the primary server are sent to the standby database directly from the log buffer of the primary server. Thus the two servers stay completely in sync with each other. There are three modes of operation for HADR; synchronous, near synchronous and asynchronous.

Highlights

Synchronous Mode

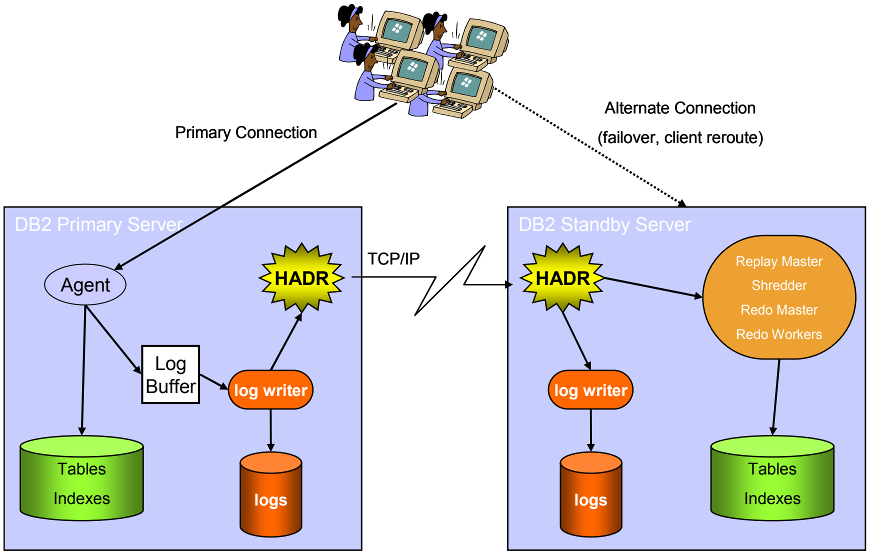
In synchronous mode, DB2 ensures that the log records being written to disk on the primary server are also written to disk on the standby server before an application receives a successful return code to their commit statements. In this mode, there is a guarantee that no committed transactions will ever be lost as both servers stay completely in sync.

Near Synchronous Mode

In near synchronous mode, DB2 ensures that the log records being writing to disk on the primary server are in memory at the standby server (but perhaps not on disk at the standby) prior to notifying an application that their commit statement was successful. In this mode, there will never be any transactions lost unless both the primary and standby fail simultaneously.

Asynchronous Mode

In asynchronous mode, DB2 will write the log buffer to disk on the primary server and ensure the log buffer has been passed down to the TPC/IP socket to be sent over to the standby. In this case, it would be possible to lose a committed transaction if the primary failed and the packets containing the log buffer did not make it to the standby server prior to a takeover.



"The DB2 automatic client reroute feature will automatically reconnect the application to the standby server so that the application can continue to function."

Automatic Client Reroute

When a client is connected to a database and that server fails, the DB2 automatic client reroute feature will automatically reconnect the application to the standby server so that the application will continue to function. Any in-flight transaction is rolled back and the application can then continue from where it left off. Automatic client reroute is configured on the database server, making mass deployments much simpler than competitor's products which require client side configuration for application failover.

When a client application connects to the database, the standby server information is pulled back to that client where it can then be used (at any point in the future even after the client disconnects) to automatically reestablish a connection to a standby server if the primary becomes unavailable.

Highlights

Failover and automation of failover

With HADR, the failover to a standby server is extremely simple. There is just a single command (or single click in the HADR monitor graphical interface) called TAKEOVER. The takeover command has two modes of operation. The first is a simple takeover in which the primary and standby switch roles. This allows for a graceful switch over in which the primary and standby coordinate with each other to change roles (primary becomes standby and standby becomes primary). This method is useful for rolling upgrades where you apply a fix or upgrade to the standby server, switch roles and then apply the fix or upgrade to the primary server.

In the event of a failure on the primary server you perform a TAKEOVER BY FORCE in which the standby server assumes the role of primary server without coordinating with the old primary (since it is likely down, it is not possible to contact the primary). When this command is executed, the new primary will replay any logs it still has in memory, undo any in-flight transactions and open the database for new transactions. Note that because the standby was only processing insert/update/delete activity, most of the recently updated data pages will still be in memory on the standby and therefore the undo phase is exceptionally fast (no I/O is likely required in order to undo in-flight transactions). In fact, the undo phase of recovery completed in just 3 seconds for the 600 user SAP test described in this paper.

“the undo phase of recovery completed in just 3 seconds for the 600 user SAP test.”

Automation of the takeover is also simplified with the use of a cluster manager like Tivoli System Automation (TSA which is shipped with DB2[(8)]), HACMP, or other cluster software. DB2 also ships with the TSA scripts required to perform a takeover in the event of a server failure. See the section below for information on configuring TSA with HADR.

“Tests with TSA and HADR running an SAP R/3 workload show that failover can happen in 11 seconds for a system simulating 600 users”

Tests with TSA and HADR running an SAP R/3 workload show that failover can happen in 11 seconds for a system simulating 600 users.

It should also be noted that HADR is not a shared disk implementation or an active/active configuration. Therefore, the standby server is able to deliver the same 100% performance (which the primary was able to deliver) after a failure occurs. Thus your business can keep running at peak performance even when a node in the cluster has failed. This is not the case with active/active configurations which typically degrade in performance when one node fails.

Highlights

"shared disk clusters require synchronization and serialization mechanism that impact their ability to scale"

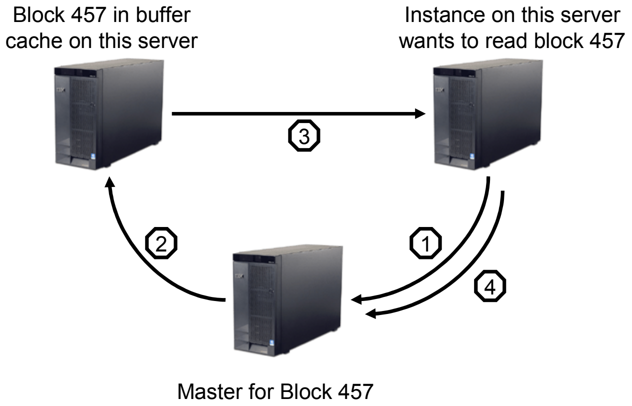
"In a four node cluster benchmark, the global cache service was consuming 12% of each server."

Comparison with Oracle Real Application Clusters (RAC)

Oracle Real Application Clusters (RAC) is a shared disk implementation of a database cluster. This means that multiple physical servers share a single database stored on disk that can be concurrently updated by all servers in the cluster. In order to avoid two servers trying to simultaneously update records on the same data page, shared disk clusters require synchronization and serialization mechanisms that impact their ability to scale and/or require the database administrator to take steps (like partitioning data and applications) in order to reduce the contention between servers in the cluster.

Each server in a RAC cluster is running a component known as the Global Cache Service (GCS) which is responsible for synchronizing and serializing access to data among the nodes in the cluster. As well, each block (or page) of data has an owning node, known as a master node. All servers in the cluster act as a master for a subset of the data blocks in the database in order to distribute the workload amongst the servers. An SAP SD Parallel benchmark run with Oracle RAC shows that the overhead of synchronization and serialization is significant [1]. In a four node cluster benchmark, the global cache service was consuming 12% of each server. Here is an example of what happens when one server needs access to a page of data that is currently in the memory (buffer cache) of another server.

1. Global Cache Service on requesting server asks master server for data block
2. GCS on the master server sends request to instance with most recent copy of the block telling it to send the block over to requesting server
3. Entire data block is sent over the network to the requester
4. Requester sends another message to the master server indicating that it now has the most current copy of the block.


Block 457 in buffer cache on this server
Instance on this server wants to read block 457
3
2
1
4
Master for Block 457

As you can see the amount of network traffic and the CPU overhead to manage this communication is non-trivial and can seriously impact the scalability and performance of a cluster.

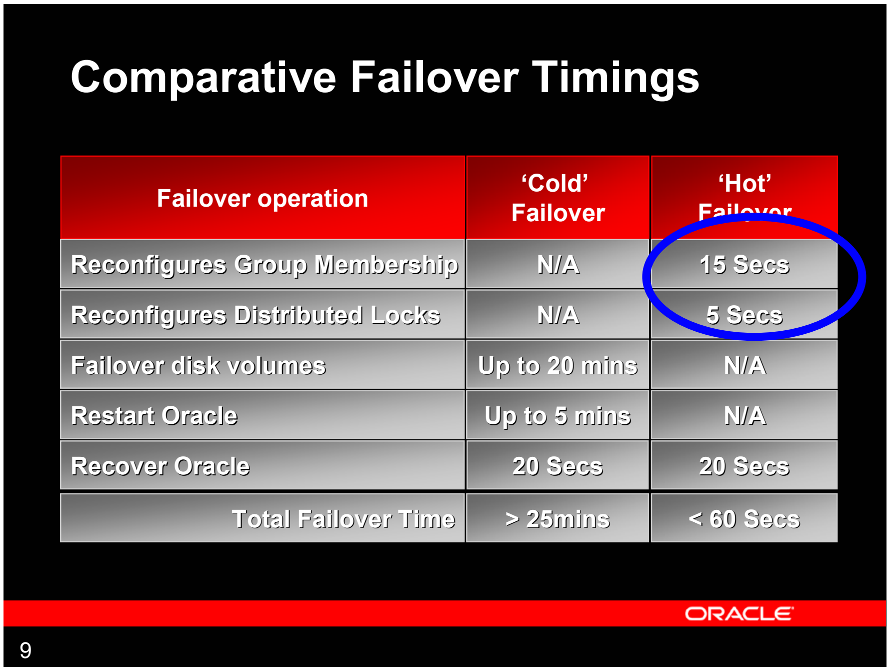
Highlights

"While this reconfiguration is happening, all servers in the cluster are frozen and your applications will hang."

"Oracle's own presentation shows that the period of the freeze is around 20 seconds "

What happens when a node fails?

If one node in the RAC cluster fails, the master node for a portion of the database has been lost in addition to any in-flight transactions running on the failed node. When a node fails, Oracle must move the ownership of data blocks off of the failed node and redistribute this ownership to the surviving nodes. This is known as Global Cache Service reconfiguration. While this reconfiguration is happening, all servers in the cluster are frozen and your applications will hang. Why? Because while this is running, GCS will not respond to any request from any node in the cluster. Remember that GCS is required if you want to read or write any page from or to disk and it is required if you want to upgrade your intention from read to update on any data page. Oracle's own presentation[2] shows that the period of the freeze is around 20 seconds (15 seconds for reconfiguring group membership and an additional 5 for reconfiguring distributed locks).



Comparative Failover Timings

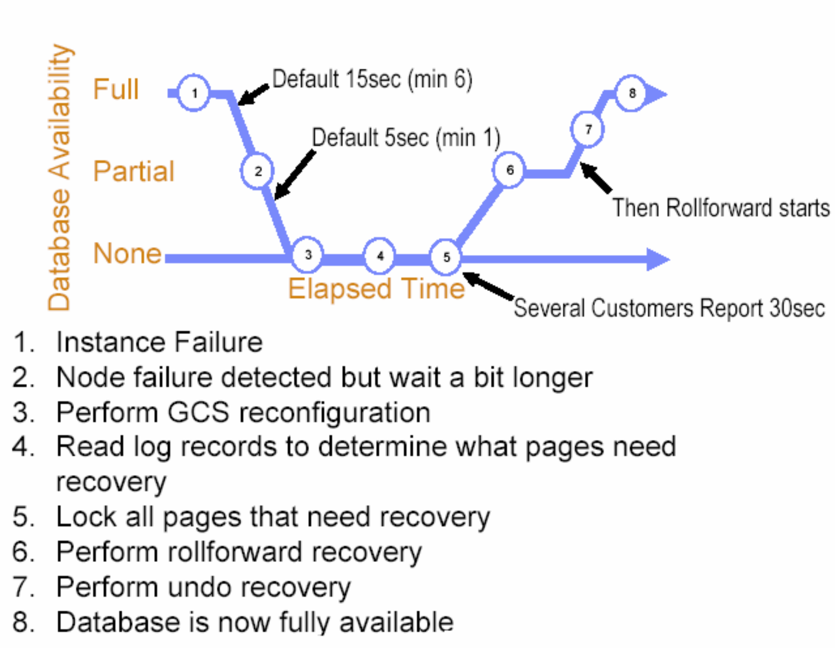
Failover operation	'Cold' Failover	'Hot' Failover
Reconfigures Group Membership	N/A	15 Secs
Reconfigures Distributed Locks	N/A	5 Secs
Failover disk volumes	Up to 20 mins	N/A
Restart Oracle	Up to 5 mins	N/A
Recover Oracle	20 Secs	20 Secs
Total Failover Time	> 25mins	< 60 Secs

ORACLE

9

This is also documented in the Oracle Real Applications Administration Guide as figure 7-1 as shown below.[3]

At point 1 a node in the cluster fails. Oracle's cluster manager will not detect a node failure until after a number of heartbeats are missed. This is typical cluster manager behaviour. With Oracle this interval is configurable as part of the cluster manager configuration (cmcfg.ora) and has a default value of 15 seconds and can be set as low as 6 seconds. Oracle then waits for another timeout period to ensure the node has really failed before ejecting it from the cluster. The reason for this is that with RAC you must ensure that a node is down before you treat it as so. If two nodes in the cluster each think the other is down when they are not, you get what is called a split brain scenario where two instances on a shared disk database are acting independently and will corrupt the database. After the timeout periods, Oracle reconfigures group membership and distributed locks and note that the vertical axis indicates **NO Database Availability** during this period



1. Instance Failure
2. Node failure detected but wait a bit longer
3. Perform GCS reconfiguration
4. Read log records to determine what pages need recovery
5. Lock all pages that need recovery
6. Perform rollforward recovery
7. Perform undo recovery
8. Database is now fully available

What happens after a node fails?

It should be noted that in an active/active cluster configuration, after one server in the cluster fails, you are running with degraded performance. For example, in a two node cluster, if one node fails, you now only have 50% of your server capacity available to run your business. Active/passive configurations are designed to be able to run your application in the event of a node failure with 100% capacity. If you need 100% performance after a failure then you must have extra capacity available prior to the failure. In the case of Oracle RAC, you would have to use a minimum of 3 servers (or 2 over sized servers) in order to survive a failure and maintain performance. Keep this fact in mind when you read the next section on the Real Costs of RAC.

"in an active/active cluster configuration, after one server in the cluster fails, you are running at degraded performance."

The Real Cost of a RAC solution

Oracle claims that you can save money by using Real Application Clusters on commodity hardware. What Oracle doesn't mention is that the price of RAC is a 50% uplift on top of Oracle Enterprise Edition. The result is that you may be able to shave 10-30% off of your hardware costs but you may end up paying up to 344% more for your software compared to a comparable DB2 solution.[(4)]

In fact, an article was published in the International Oracle Users Group Select Journal (3rd Qtr. 2003) entitled "You Probably Don't Need RAC". The conclusion of the Oracle User Group article states, *"Most likely, you probably don't need RAC. Alternatives will usually be cheaper, easier to manage and quite sufficient."*

"an article was published in the International Oracle Users Group Select Journal (3rd Qtr. 2003) entitled 'You Probably Don't Need RAC'"

The article goes on to say *"Consider Larry's vision of cheap Intel-based Linux clusters. For instance, let's buy those two cheap,* *4-cpu Intel boxes and put them together in a cluster with Oracle9i and RAC on top:*

· Price for the hardware: About US $15,000

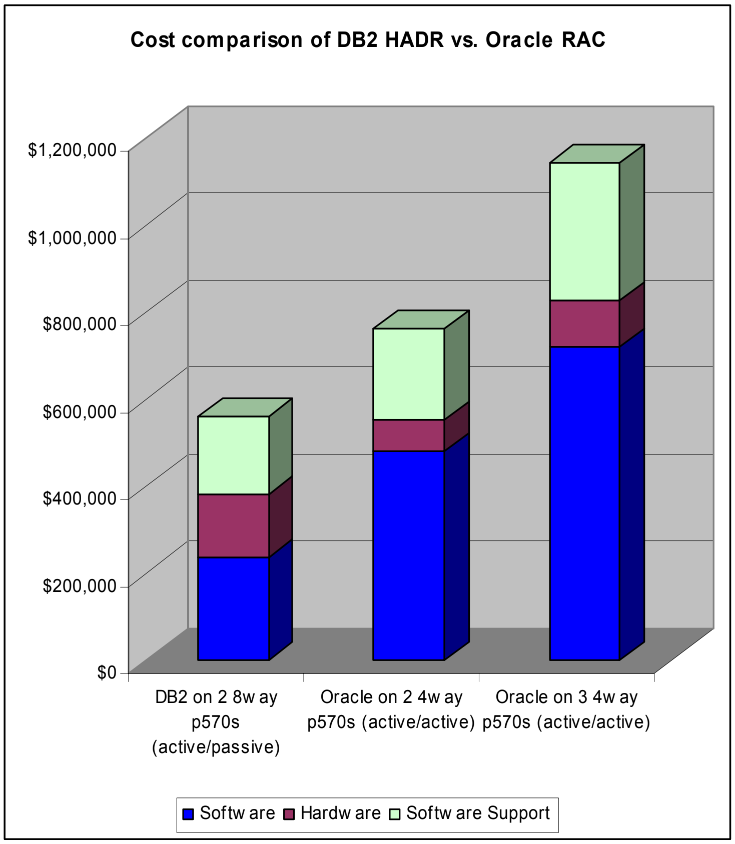
· Price for the OS (Linux): About US $50

· Price for Oracle w/ RAC: US $480,000,

To sum up, that adds up to $500,000. Moreover, it's one dollar given to the box movers for every $32 Oracle receives."

"There are other indirect costs associated with implementing RAC. First, your personnel must be more skilled, with respect to RAC and clusters. Second, you'll have to consider the availability of a development environment (and possibly a test environment) that consists of both a cluster and RAC"

Below is a comparison between an active/passive DB2 cluster running HADR with an active/active cluster running Oracle RAC. Using identical server processors with the same number of "active" CPUs, the DB2 solution costs you less.



The above comparison shows DB2 ESE running on two 8way p5 570s in an active/passive configuration compared to Oracle EE plus RAC running on two 4way p5 570s (middle bar)). In this comparison, if the DB2 active server fails, you have a standby server capable of handling 100% of the throughput from the primary. In the Oracle RAC case, if one of the 4ways fails, you now only have 50% of the capacity to continue to run your business. To mitigate this problem, you would actually need to purchase three 4way servers to have an equivalent HA solution to DB2.

Highlights

"There are other indirect costs associated with implementing RAC. First, your personnel must be more skilled, with respect to RAC and clusters. Second, you'll have to consider the availability of a development environment (and possibly a test environment) that consists of both a cluster and RAC"

The price for this configuration is represented by the right hand bar above and shows that the Oracle solution is 2x more costly than the DB2 solution even when using smaller SMPs in an active/active RAC configuration. Note that with DB2, you only pay for 1 cpu on the passive server regardless of the number of CPUs on that server. Of course you can use the passive server for a number of other tasks (like development, test, staging data, etc).

As well, HADR works at the database level, so if you have multiple databases, you could put database A on server 1 with its standby on server 2 and have database B on server 2 with its standby on server 1. Thus both machines are active and both are standbys.

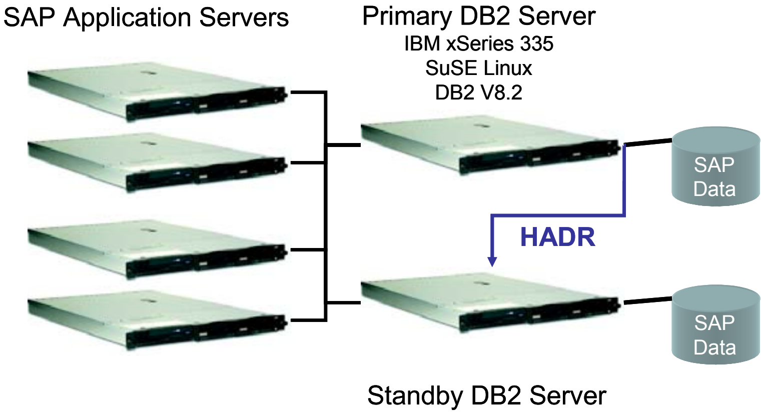
All prices are list price from the IBM and Oracle websites and include servers, software and one year of software support.[5]

Highlights

"The price … shows that the Oracle solution is 2x more costly than the DB2 solution even when using smaller SMPs in an active/active RAC configuration."

Delivering under 15 second failover with DB2 for Real Applications

For a real world availability test, SAP R/3 Sales and Distribution was chosen as the application to simulate a large concurrent user workload[6]. The configuration included two IBM eServer x335s, each with 4 Intel Xeon processors running SuSE Linux and DB2 V8.2. To manage and automate the failover, IBM Tivoli System Automation (which ships with DB2 on Linux) was used. Setting up this configuration is significantly simpler than setting up a RAC cluster. Below is the environment deployed followed by the setup and configuration steps used for this test.



It is assumed that the hardware is available and that DB2 has been installed. There are then two additional steps required to enable the solution:

1. Configure HADR
2. Tivoli System Automation setup and configuration

Highlights

"The failover test was performed by simulating 600 SAP users running on a set of SAP application servers"

Configure HADR

The following are the steps required to setup HADR between two servers. These steps can be performed manually or you can simplify the setup by using the HADR Configuration Wizard which performs all of the following tasks on your behalf by asking you just 5 questions.

1. identify primary and standby servers and instances
2. backup primary database and restore on standby
3. specify the TCP/IP ports for HADR to communicate with
4. enable client reroute
5. determine the synchronization mode

After these 5 steps you simply start HADR on the standby and primary with the following commands respectively

```
db2 start hadr on database db_name as standby
db2 start hadr on database db_name as primary
```

This will start the HADR processes and bring the standby database in sync with the primary.

Tivoli System Automation setup and configuration

Configuring TSA requires only two steps:

1. Install IBM Tivoli System Automation for Multiplatforms
2. Setup TSA to manage the DB2 instances

Install IBM Tivoli System Automation for Multiplatforms

TSA must be installed on both machines locally. To install TSA follow the installation procedure described in IBM System Automation for Multiplatforms, Guide and Reference.

Setup IBM Tivoli System Automation to manage the DB2 instances

Now that all the components are installed, the final step is to have TSA manage the DB2 instances and the HADR pair. Simply:

Create the cluster domain:

```
mkrpdomain hadr_domain server1 server2
```

Activate the newly created domain.

```
startrpdomain hadr_domain
```

Run a script (which ships with DB2) to register the two DB2 instances with the cluster manager and tells the cluster manager to run the TAKEOVER command if one node fails

```
reghadrsalin -a inst1_name -b inst2_name -d db_name
```

That's it! Now the HADR pair is controlled by the cluster manager. If the primary server fails, TSA will automatically trigger the takeover. Compare this configuration to the 422 page Oracle RAC Installation and Configuration Guide.

Test Results

The failover test was performed by simulating 600 SAP users running on a set of SAP application servers. When the system was fully loaded with transactions generated by 600 concurrent users, the primary server was powered off. The HADR resource group automatically failed over to the standby node, resulting in the standby instance turning into the HADR Primary for the database. When the original primary server comes back online, you can reintegrate it into the HADR cluster by issuing the following command:

```
        db2 start hadr on db hadrdb as standby
```

The HADR pair will now be re-established.

Failover Timings

The first test involved powering down the machine hosting the HADR primary database. The time from the power off until the first application was able to successfully complete a transaction on the standby was 11 seconds. Of this 11 seconds, nearly 8 was required by the cluster manager to detect the failure and initiate the DB2 HADR Takeover process. The remaining time was used by the HADR Takeover and the rolling back of in-flight transactions.

"The time from the power off until the first application was able to successfully complete a transaction on the standby was 11 seconds."

A controlled failover was also tested, where the primary and standby switched roles via the HADR commands. This scenario is useful for applying rolling upgrades and/or to test your failover scenarios. In this test the recorded time was just under 13 seconds. The added 2 seconds was due to the standby and primary communicating in order to coordinate the controlled takeover.

As you can see the setup is quite simple for the availability levels that can be achieved with this configuration.

Conclusions

Highly availability has always been a critical success factor for mission critical applications. Some vendors have decided that your need can be their ticket to increased revenue by charging you a premium price for clustering software. DB2 UDB V8.2 Enterprise Server Edition includes the High Availability Disaster Recovery (HADR)feature which has been shown above to deliver the same levels of availability as Oracle RAC for significantly less cost.

"Don't overpay for your high availability solution… a DB2 active/passive solution will save you money while delivering equal or better availability"

Don't overpay for your high availability solution. Make sure you include hardware **and** software costs (including support and upgrade subscription) in your price comparisons and you will likely find a DB2 active/passive solution will save you money while delivering equal or better availability. In addition, an active/passive configuration is easier to manage, supports existing applications, supports existing development and administration tools and is certified for ISV applications like SAP, PeopleSoft, Siebel and others on all supported operating systems.

Contact your IBM representative to learn more about increasing the availability of your database system without paying too much.

.

References

1. All references to SAP R/3 SD-Parallel Standard Application Benchmark results are accurate as of November 9, 2004.

   These benchmarks fully comply with the SAP Benchmark Council's issued benchmark regulations and have been audited and certified by SAP.

   More information is available under http://www.sap.com/benchmark.

   - SAP SD Parallel, 12,000 SD Parallel user, 1,208,330 Order Line Items / Hour, 1.92 Sec. Avg. Dialog Response Time, Certification Nr. 2002031, SAP R/3 4.6C, Oracle9i Real Application Clusters (RAC), HP AlphaServer ES45 Model 2, HP Tru64 Unix V5.1, Alpha EV6.8CB (21264C) 1000Mhz, 2nd Level / 8MB, 4 active nodes, 4 CPUs per node, 32768 MB memory per node.
   - SAP SD Parallel, 6,580 SD Parallel user, 668,330 Order Line Items / Hour, 1.81 Sec. Avg. Dialog Response Time, Certification Nr. 2002030, SAP R/3 4.6C, Oracle9i Real Application Clusters RAC), HP AlphaServer ES45 Model 2, HP Tru64 Unix V5.1, Alpha EV6.8CB (21264C) 1000Mhz, 2nd Level / 8MB, 2 active nodes, 4 CPUs per node, 32768 MB memory per node.
   - SAP SD Parallel, 3,640 SD Parallel user, 404,000 Order Line Items / Hour, 0.81 Sec. Avg. Dialog Response Time, Certification Nr. 2002029, SAP R/3 4.6C, Oracle9i Real Application Clusters (RAC), HP AlphaServer ES45 Model 2, HP Tru64 Unix V5.1, Alpha EV6.8CB (21264C) 1000Mhz, 2nd Level / 8MB, 1 active node, 1 passive node, 4 CPUs per node, 32768 MB memory per node.
   - To obtain the full submission data for the Oracle benchmarks, you can contact SAP.
2. Presentation given by Marshall Presser, Principal Technologist Oracle Corporation to the Beowulf users group, 11 May, 2004. http://www.bwbug.org/docs/BWBUG-May2004.ppt
3. Oracle 9i R2 Real Application Clusters Administration Guide. Page 7-12, Figure 7.1
4. Based on DB2 WSE + HADR option running on 2 4way OpenPower 720 server with Linux vs Oracle EE + RAC on 3 4way OpenPower 720 servers running Linux. Note 3 servers are required for RAC to deliver 100% capacity in the event of one node failure as is the ability in the DB2 solution. Total reference list price for the DB2 solution based on web based pricing including 3yrs of software support is $160k vs the Oracle web based pricing including 3yrs of software support of $552k. All prices are list prices based on www.ibm.com, www.oracle.com, www.sun.com as of November 9, 2004 and are subject to change without notice.
5. Prices are current as of November 9, 2004, exclude applicable taxes, and are subject to change by IBM without notice
6. The SAP R/3 Sales and Distribution workload was not an SAP Benchmark or an SAP endorsed test. Rather the SAP workload was used to simulate a real application scenario.
7. Oracle 10g Real Application Clusters Installation and Configuration Guide Part No B10766-07 http://download-west.oracle.com/docs/pdf/B10766_07.pdf
8. IBM Tivoli System Automation for Linux is included with DB2 UDB v8.2 at no additional cost for use by DB2 products only in a 2 node cluster configuration.

.

IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in United States of America
10-2004

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The information in this white paper is provided AS IS without warranty.
Such information was obtained from publicly available sources, is current as of 08/26/2003, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.

.

e business software