January 2006

Database Resiliency

Calvin X. Chu
Worcester Polytechnic Institute

Venera Varbanova
Worcester Polytechnic Institute

Project Number: MQP MXC WO51

Global Resiliency in Investment Banking Technology: Database Corruption at JP Morgan Chase

A Major Qualifying Project Report
Submitted to the Faculty of
Worcester Polytechnic Institute
In partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted By:

Calvin Chu

Venera Varbanova

Arthur Gerstenfeld

Michael Ciaraldi

Project Center:

Wall St. New York, NY
B Term 2005

Sponsoring Agency:

JP Morgan Chase

Submitted To:

Project Advisors:

Michael Ciaraldi
Arthur Gerstenfeld

On-Site Liaison:

Tom McLean

Date: December 13, 2005

Abstract

Our project was sponsored by Global Resiliency in Investment Banking Technology at JP Morgan Chase. We investigated a number of database corruption issues, which had had a significant financial impact upon the firm. Working from industry research, interviews with managers and database administrators, and internal problem data, we produced a research paper, describing the problems, featured solutions, and the trade-offs that are involved with implementing those solutions. A cost-benefit analysis was performed to justify investing in the recommended solutions.

Executive Summary

One of the greatest differences between business in the past and business now is the requirement of high availability and business continuity. Obviously, JP Morgan follows these requirements. In fact, they devote large amounts of money to keep critical applications up and running. These applications cannot afford to have a minute of downtime, or the financial impact would be overwhelming, affecting the financial health of the company, and potentially the world economy.

A database is usually described as corrupt, or damaged, if, when trying to extract or modify some information, errors appear and/or the information to be extracted turns out to be lost, incomplete or incorrect due to software or hardware fault. This is called database corruption. Database corruption can bring down systems for hours, even days, thus crippling the business. Yet because of its infrequency, seldom do people see database corruption as a threat. Those who have experienced it all agree that the financial impact is staggering. Considering that fact, businesses must plan for such an occasion. What can be done so that database corruption can be prevented and more importantly, how can it be mitigated so there is minimal amount of data loss and latency, when it occurs?

Most feel prevention is difficult as new, unknown threats to infrastructure will always exist, but measures can be taken to alleviate the effects and to recover faster after it happens. Developing an infrastructure that is prepared for database corruption at all times will prevent JP Morgan from not only negative impacts financially, but also from negative impacts to goodwill and their reputation.

Our method in attaining information about this problem was to look at internal and external documents about the database corruption, specifically past problem tickets internally. Externally, we looked at white papers for various products specializing in high availability and industry websites. These sources were very helpful, but the most useful resource we had was the people within the organization. We conducted interviews and electronically corresponded with a number of different people about database corruption. We found names of people through networking and through searches online for names of people involved with specific situations or applications. It was difficult to contact some people and in those instances we had some of our established contacts try to get in touch with them to set up lines of communication with us.

From all of our interviews, most respondents did not see database corruption as an issue and saw database outages as inevitable. Thus, they did not see various products such as Sybase Mirror Activator, Golden Gate and Oracle Recovery Manager as something with much value.

While we disagree that these products are of no value, we cannot advise adopting technology that is too new or too expensive. For instance, as of right now, we do not advise the adoption of Sybase Mirror Activator. JP Morgan has a history of being adopters later on during the product's life cycle and for good reason. They deliberately do this because they allow others to adopt early and run into any bugs, and they get fixed before JP Morgan purchases. As Mirror Activator is still very new and there are only a few adopters, it would be unwise to pay the high cost to implement this technology.

We have found a number of best practices, including but not limited to: more automated monitoring and archiving and more redundancy, like snapshots and hot stand-bys.

Automated monitoring and archiving

We found that many of the problems with database outages could have been prevented with better monitoring or could have recovered faster had there been a log of transactions carried out. While this solution seems obvious, not all systems have implemented techniques like this, even some critical ones. Implementing automation removes the weight off the shoulders of people who need to monitor the systems and also could pick up what a person cannot.

Redundancy

Ideally, an organization would like to have as many up-to-date copies of their data as possible. This seems redundant, but that's the point. The problem is how to keep the data as up-to-date as possible, while minimizing cost of storage, bandwidth usage, etc. A low-cost solution would be taking snapshots of file systems. A point-in-time copy of a file system or storage volume would be made and used along side the logs to recover faster. Also, a good approach would be to have hot standby systems in place. In a failover situation, time is critical. The difference in time to failover from having a hot stand-by opposed to a cold one could save an organization thousands of dollars. These are just a couple of possible solutions to mitigate the effects of database corruption.

Acknowledgements

Firstly, we would like to express our gratitude to JP Morgan Chase and Worcester Polytechnic Institute for allowing us to take part in such a coveted program. There are many people who deserve recognition for their contributions to our project and unfortunately and obviously, we cannot list them all, but we can name a few who stepped above and beyond to secure a successful experience for all. We would like to thank Tom McLean, Marisa Giliberti, Margarita Ramirez, Vicky Sadoff, John Storm, Karen Hengerer and the rest of the JP Morgan staff members that were involved.

We would also like to recognize the efforts of our advisors, Professor Art Gerstenfeld and Professor Michael Ciaraldi. Their efforts, combined with the projects program, embody the heart and soul of what higher education is supposed to be: a bridge to the real-world. This program allowed us to broaden the knowledge acquired in a classroom and apply it beyond the chalkboard. For that experience, we are indebted.

Table of Contents

Table of Figures

1. Introduction

High availability, business continuity, and disaster recovery; terms heard over and over when discussing business. They may seem like buzz words, but they represent a movement. This movement aims to replace the concept of "business 24/7" with "business ∞". Simply mentioning these phrases could grab the attention of the room, as everyone wants to know the secret to obtaining perpetual commerce. Currently, no system is 100% fail proof and constantly available; hardware and software fail, corruptions occur and people make errors. The closest systems we have to perfection are the ones that are prepared for such outages and even have strategies for when any unknown or uncommon problems arise. Database corruption is an example of these problems.

Database corruption is something that can bring down systems for hours, even days, thus crippling the business. Those who have had experience dealing with it agree that it is a major issue and the financial impact is staggering. Yet because of its infrequency, seldom do people see database corruption as a threat and do not spend time and resources to address it. As the saying goes, "You can't catch what you can't see." But, how true is that saying? You could study the actions of your prey to learn about it and discover trends. It may be difficult to prevent it from entering your domain, but you could set up traps to catch it when it invades your territory. All this can be done without ever seeing it coming. The same is true for database corruption.

The purpose of this document is to discuss, at a high level, what best practices are being implemented externally to obtain maximum availability, minimal latency in failover and no data loss, with respects to database corruption. This document will also take a brief look at JP Morgan's history of dealing with such instances and what it is

doing to minimize impact of future occurrences, including potential third party solutions being analyzed. With these findings, the Global Resiliency Group aim to improve the reliability and resiliency of the internal databases systems.

2. Background

2.1. Industry overview

The requirements of business today are completely different than those of the past. One of the greatest differences is the requirement of high availability and business continuity. High availability, according to DM Review Magazine, is "protocol and associated execution that ensures a certain relative degree of computing-system operational continuity in any downtime event." Also, the magazine defines business continuity as "the degree to which an organization may achieve uninterrupted stability of systems and operational procedures."[1]

As illustrated in Figure 1, the modern concepts of disaster recovery and business continuity were hatched around the beginning of the 1990s mainly in the form of legal requirements for banks and financial institutions to set up contingency plans in the event of a disaster.

[1] www.dmreview.com/resources/glossary.cfm?keywordId=ALL

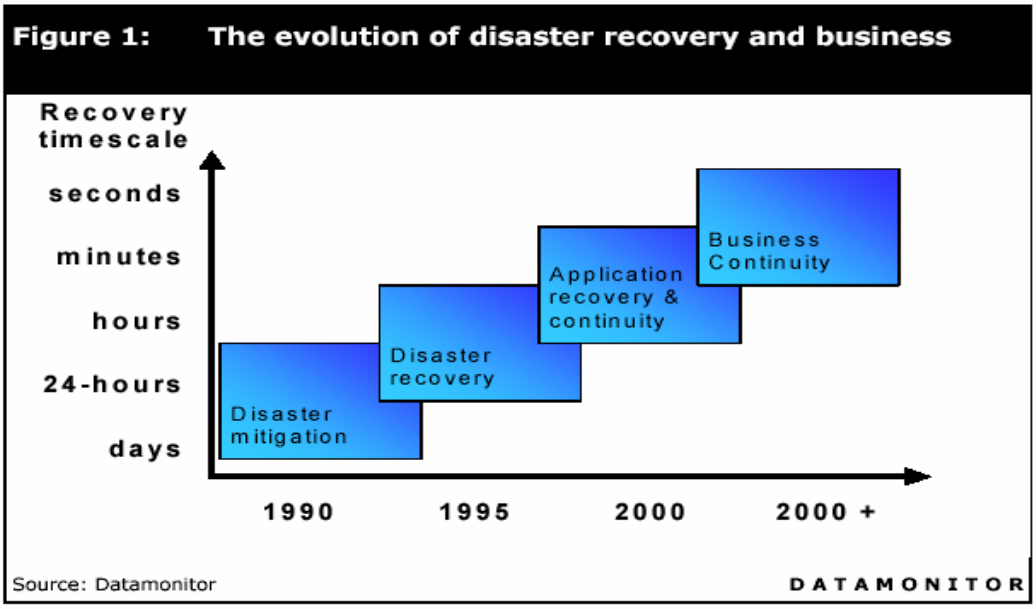

Figure 1: Evolution of Disaster Recover and Business[2]

2.1.1. Government regulations

According to the Gartner Group, businesses in the finance industry have special requirements for their business continuity. Some of those requirements that impact business continuity/ disaster recovery are displayed in figure 2.

[2] www.sybase.com...BCPDRv1.22May02.pdf

Industry Sector	Significant Laws and Regulations	Impact on BCP	Comments
Finance	Federal Financial Institutions Examination Council (FFIEC) Handbook, 2003-2004 (Chapter 10)	Specifies that directors and managers are accountable for organizationwide contingency planning and for "timely resumption of operations in the event of a disaster."	This chapter — on an operational level — supplants many other BCP guidelines. It covers examination requirements for all companies regulated by the Federal Deposit Insurance Corp. (FDIC), Federal Reserve Bank (FRB), Treasury Department, U.S. Office of the Comptroller of the Currency (OCC), Office of Thrift Supervision (OTS) and National Credit Union Administration (NCUA).
	Basel II, Basel Committee on Banking Supervision, Sound Practices for Management and Supervision, 2003	Requires that banks put in place BC and DR plans to ensure continuous operation and to limit losses.	After 2007, influence of Basel II will be limited to about 30 U.S. banks but will spread as a best practice via "audit creep."
	Interagency Paper on Sound Practices to Strengthen the Resilience of the U.S. Financial System, 2003	More focused on systemic risk than individual enterprise recovery. Requires BCPs to be upgraded and tested to incorporate risks discovered as a result of the World Trade Center disaster.	Influences companies that are regulated by Securities and Exchange Commission (SEC), OCC and Board of Governors of the Federal Reserve System (FRS). Authorizes the OCC to take action against banks that fail to comply with requirements for DR by the U.S. financial system.
	Expedited Funds Availability (EFA) Act, 1989	Requires federally chartered financial institutions to have a demonstrable BCP to ensure prompt availability of funds.	

Figure 2: Government Regulations for Finance Industry[3]

[3] *www.gartner.com...laws_influence_business_cont_128123.pdf*

As time progressed and technology improved, business became more and more dependant on networks, thus requirements became stricter. As the dependence grew stronger, it became in an organization's best interest to become fully available to gain and maintain market strength, since it was common sense that the most reliable companies would be known as industry leaders. It was no longer good business protocol just to keep up with the requirements, but rather to surpass them.

2.2. JP Morgan Chase Overview

2.2.1. Early Years

JP Morgan has a rich and deeply rooted history spanning back to late 18th and early 19th century with the Manhattan Company. It started as a water company, but one of its founders, Aaron Burr, had created clauses in their charter that allowed them to act in some ways, as a bank. Some historians believe this was done deliberately to break up the federalist banking monopoly. The company's other founder, Alexander Hamilton, did not agree with this clause and once the charter was approved, he cut ties with it. As the story goes, the animosity between the two men grew and Burr challenged Hamilton to their famous duel, where Hamilton died.

2.2.2. JP Morgan in the 1800s

More direct links to the current company formed in the mid-to-late 1800s. In 1871, J. Pierpont Morgan, Sr. and Anthony Drexel, a Philadelphia banker, formed a private banking partnership called Drexel, Morgan & Co. As the century was coming to an end, two of the firm's senior partners passed away, J. Pierpont's father, Junius Morgan (1890), and Anthony Drexel (1893). At that point, J. Pierpont Morgan, Sr. consolidated

and reorganized his family's private banking interests and became senior partner. The New York firm was renamed to J.P. Morgan & Co.

Around the same time of the partnership of Morgan and Drexel, John Thompson, a banker from New York, founded Chase National bank in 1877. The bank was named in honor of friend and former Secretary of the Treasury, Salmon P. Chase.

2.2.3. JP Morgan in the 1900s

Over the next 50 years, the country had continued to grow economically strong, but in 1929, the stock market crashed. This same year, two Columbus, Ohio banks, Commercial National Bank and City National Bank of Commerce, merged to form City National Bank and Trust Co. which would eventually become BankOne. In 1955, Chase National Bank and The Bank of the Manhattan Company merged, forming Chase Manhattan Bank. Chase, known for its strength in international, corporate, and correspondent banking was a perfect combination with The Bank of The Manhattan Company's strength in branches and retail banking.

2.2.4. JP Morgan Today

These three banks, JP Morgan & Co., Chase Manhattan, and BankOne, grew to be strong financial institutions nationally and internationally and over the next 50 years, grew to be even greater. At the turn of the millennium, J.P. Morgan & Co. merged with The Chase Manhattan Corporation and was renamed to J.P. Morgan Chase and Co. Four years later, J.P. Morgan Chase and Co. merged with Bank One Corporation.[4]

[4] http://jpmc-intranet.bankone.net/corphistory/timeline.asp

2.2.4.1. Global Resiliency Group

At JP Morgan, the Global Resiliency group, headed by Gloria Guman and Tom McLean, works alongside the Production Assurance group, headed by Karen Hengerer and Steve Donnelly. Both are managed by John Storm. The resiliency group was designed to "identify, prioritize and address weaknesses across all dimensions and implement sustainable improvements."[5]

The Resiliency group follows a guideline called the business process index (BPI) and their targets for that index for 2005 are Q1-99.6%, Q2-99.7%, Q3-99.9%, and Q4-99.9%. BPI is calculated by taking the hours of uptime and dividing it by the total hours output. The previous year, they had their best week during the week of December 26th, where BPI was 99.78%. Conversely, they had their worst week during the week of March 14th. This was due to the M3 outage.[6] They have come to some major conclusions about the organization and those insights, in their own words are listed below.

- Management Disciplines need to be rebuilt quickly:
    - Change
    - Crisis
    - Problems
    - Network
- BCP needs to evolve to a more incrementally resilient approach
- Ownership, Accountability & Urgency Culture must be incubated
- Facilities, space, power, storage capacity is at critical levels[5]

[5] Internal Document: 01_Resiliency_Story-Summary_Jan2004_John_Storm.pdf
[6] Internal Document: Roadmap to 3 9s.ppt

What this means is that they need to manage their operations better, focusing on managing change, during crisis, managing problems and keep their networks in ideal conditions. As this is a critical element to the success of their operations, they need to rebuild them as soon as possible.

The second point states how they need to have a constant evolution in their business continuity planning. Rather than have one major change every so often, they need to have many small changes almost constantly, thus an adaptive environment that is more resilient.

The third point states how the environment they exist in must be one that is filled with people who take responsibility for their work, regardless of the situation, good or bad. Also, it is an environment where everyone is looking out for each other in a positive way so that everyone is kept accountable, without ever pointing fingers and when crises occur, everyone is focusing on solutions not blame.

The final point is about how they must keep their physical environments so that critical applications can exist with minimal fear of failure. This is a major focus of the rest of this document, but this environment includes strong maintenance of facilities and good monitoring of processes.

3. Methodology

There were three main ways of accumulating information, for this project: industry research, internal problem data mining, and interviews.

3.1. Industry research

Searches for trends within the industry were done throughout the entirety of the project, but the bulk of the industry research was done at the beginning of our time at JP Morgan. The research outside of the company was mostly from internet research, industry white papers and magazines. This information not only proved quite useful as it provided us with guides for what to ask in our interviews, but also it familiarized us with the terminology used in this industry. The focus was to look at related DBMS websites along with literature and periodicals that had pertinent information, such as *Computer World Magazine*. While this was definitely what helped us prepare for our interactions with people at JP Morgan, many conversations led to better research topics, for instance, what companies and products the company recommends and doesn't recommend. This information was certainly useful, but the most useful pieces of information provided were the problem data they found for us.

3.2. JP Morgan problem data research

Finding problem data was difficult as one would have to dig quite deep to find something relevant and detailed enough for analysis. At first, we did not know where to go for data, but as time progressed, we made a number of connections that helped us by searching for what we were looking for. We received a number of different types of data. They took forms of trouble ticket reports, summaries of outages, or even phone calls.

The trouble tickets we got were fairly detailed, but few were pertinent and detailed enough. There were a few that proved very useful, but we had only obtained them after a few weeks of searching and had to find specific people who were directly involved with the incidents. The issue summaries we received were not too helpful, but did help us confirm the infrequency of database corruptions and figure out best practices. We also found contacts through these summaries.

The numerous conference calls we had were useful, but rarely did people deal with database corruption. The one major conference call that seemed to open up a flow of information was with Karen Hengerer and Miguel Rivera. Miguel had solved a major outage, (see section 4.1: M3) in the IB in 2004 and Karen was our bridge to him. This conversation was one of the missing puzzle pieces to the big picture of the M3 outage and with it we knew better questions to ask and directions to take conversations and interviews. As stated earlier, this conversation was not the only useful conversation. Many interviews were conducted and from them we received the bulk of our knowledge.

3.3. JP Morgan interviews

According to IBM.com, a major aspect of solving problems is determining the problem itself. It is also states that there are specific steps to be taken to be fully established. Those steps are divided into five questions:

- What are the symptoms?
- Who or what is reporting the problem?
- What are the error codes and error messages?
- How does it fail?
- What is the business impact? [7]

The purpose of these questions is to begin dialogue about the problem at hand. The point of this process is to stress what seem to be minor factors because they could actually be major or part of a chain of minor factors that triggers the problem. These questions served as a guideline for how we interviewed people at JP Morgan.

When we conducted our first interview, we found ourselves rather overwhelmed since the person we interviewed was not only a very technical expert, but also spoke rather quickly. After this interview we decided to record all of our interviews in MP3 format along with written notes to accompany the audio files. This was the best way to handle the highly technical interviews for it allowed us to revisit the interviews after they had been conducted, without having to wait for the person to be available again to review with us. As we completed more interviews, the more we had to bring to our next interviews. For instance, once we learned of certain concepts from one DBA, we would

[7] Introduction to problem determination – (accessed Oct 20, 2005) (http://publib.boulder.ibm.com/infocenter/db2help/index.jsp?topic=/com.ibm.db2.udb.pd.doc/pd/c0011733.htm)

ask other DBA's for their opinions on those concepts. This also proved useful when considering technology purchases, like Mirror Activator, and asking others about whether they felt those products were worth the cost or could do what they claimed.

4. Motivation/Problem cases

4.1 M3 Outage

M3 is the "primary record for the Credit Derivatives Investment Grade and High Yield businesses [which] provides trade capture, valuation and risk management support to the front and middle office"[8] and in March 2004, it went offline for three days. According to an internal document, on March 10th the NYDBA escalated an issue to the JPMC AD operate department about a cache corruption. AD operate teams are assigned when problems such as these occur. Their purpose is to fix the issues. The NYDBA suggested rebooting the server and clearing out the cache and dbcc table. The severity 1 incident was announced to all M3 users and AD Operate prepared for hot backup and restored from Wednesday's backup. It came to light that Wednesday's backup was also corrupted and AD Operate requested that Tuesday morning's backup be used for the restore.

The restore from Tuesday's backup was successful and AD and operations spent the entire Thursday feeding logged trades from Tuesday and Wednesday from eblotter to M3 and ran EOD batches for each region. STS, however, encountered infrastructure problems which caused more outages and backlog. The course of action was to continue running EODs until Friday SOD and then all transactions will be sent. Then, operations overlaid Thursday's environment with Friday's environment. The AD Operate quickly

[8] Internal website: – M3

escalated to market environment CBT and worked with developers to come to a solution. Cleanup of the environment took several hours running over to Saturday. Over the next two days the transactions were continually processed and systems were restored.

During this 72-hour outage, the financial impact was estimated to be around $1 million. According to an internal document, the business was officially impacted in the following manner:

- Trades and positions were not fed to the downstream infrastructure for 3 days.
- JPMC's ability to draft confirmations, clear backlog and fax queues was severely impaired. MIS used to manage the Confirms area was not available. DTCC matching was also impacted.
- Risk Management reports were not generated automatically.
- Official CDS trading P&L was not calculated for 3 days.
- Collateral Management, Client Valuations, and Corporate Market Risk reporting teams did not receive up-to-date M3 feeds.
- During that period traders managed positions, P&L and risk via tactical tools hence reducing the overall risk of losses, financial or otherwise.[9]

[9] Internal Document: Archive Database Incident Report – 7/11/01

4.2 Trevor Outage

In May 2005, there was a major outage in the Trevor database. On Monday morning (GMT) May 2nd, the Trevor RAD team was running morning checks and came across a MySQL table corruption. At that point they began the normal procedures for repairing tables, but the repairs failed. After numerous tries, they ran the full overnight database repair. Tuesday morning it was discovered that the repair had failed. To fix this they attempted to carry-out a file restore procedure from the backup taken on the 28th and 29th of April. Normal repairs failed again so, they ran another full overnight repair.

On Wednesday morning, the team saw that the overnight repair failed again. They decided to completely recreate the table, but after repopulating data onto the database, were found that they were still unsuccessful. Finally, they decided to recreate the entire database on a different hardware server and the issue was resolved. On Thursday, Front Office switched to the new database server and risk positions could be viewed again. Throughout the entire outage, the London market was down for about 36 hours and the New York market was down for about 54 hours.

The unreliability of MySQL and the lack of an operate team were the root causes of this outage. MySQL is not recommended by the bank because it cannot handle large amounts of data and it usually causes significant fragmentation. The lack of RAD apps support created a heavy reliance on AD for communication to users in and for remediation and development. The solutions to these issues were migrating from MySQL to Sybase and an Operate Desk Aligned team was introduced to CH. This incident was declared as severity 1, as risk positions could not be viewed for three days. [10]

[10] Internal Document: Operate – Trevor Outage

4.3 TM Outage

In September 2004, there was an outage in the TM database. On Tuesday September 21st, a scheduled extract for credit data from TM, for dispatch to Typhon for reconciliation, began. This process normally takes five hours. TM servers were taken down to apply an emergency NBR, required by Tokyo JPMSL to update static data for Asia secrecy. Install was successful, TM restarted normally and systems seemed back to normal.

Early Wednesday morning, while the STS TM/Credit extract transaction was still open, the file system was filling up and Jupiter Sybase DBA's were paged. Because the transaction was still processing, Sybase could not commit the transaction to the database so the transaction log file could not be cleared. Simultaneously, other processes were adding to the log file, thus bringing it closer to its threshold. At 5:00am, the transaction logged filled up to 50 million rows beyond its limit. Overnight support was unable to stop the processes and the database server failed because of the filled transaction log. TM was shutdown and restarted.

Usual estimated rollback time, 2-3 hours, was far below system predicted estimate, which was 60 hours. Decision to roll forward on backup in PSup was cancelled by the DBA team, but decision to roll forward to F15 UAT environment was made and roll back on production with larger log files was restarted. By 4:45pm TM was ready but script testing still needed to be done for non-TRFE deals. Finally at 11:00pm TM was almost completely caught up, with about 2800 messages left in queue. When TM was

brought down, a TM report started to run for a long time and the log file filled up. Also, Sybase thresholds did not fire correctly so no DBA's were paged to manage the file. [11]

[11] Internal Document: TM outage.ppt

5. Analysis

5.1 Definition of database corruption

A database is usually described as corrupt, or damaged, if, when trying to extract or modify some information, errors appear and/or the information to be extracted turns out to be lost, incomplete or incorrect. There are cases when database corruption is hidden and can only be found by testing with special tools. However there are also very well visible instances of database corruptions, when it is impossible to connect to the database, when applications send errors to the clients (without any data manipulation having occurred), or when it is impossible to restore the database from a backup copy.[12] The latter case is the most devastating one, and it will be looked into carefully later in the paper.

5.2 Causes of database corruption

Unfortunately, there are many causes of database corruption. They vary from human error to hardware failure, from bugs in the database software to bugs in the home-grown applications that are interacting with the database. As technology evolves and gets ever more complex, the causes of database corruption grow in number. Being familiar with the causes of database corruption is essential, in order to be able to monitor and prevent problems from happening. While it is true that the causes of database corruption evolve and are in a sense impossible to completely account for, the most common causes have historically been the same. In the next few sections we are going to examine them in more detail.

[12] www.ibexpert.info/documentation/Database/DatabaseCorruption accessed Nov 2005

5.2.1. Human error

Surveys have shown that human error is one of the most common causes of database corruption. Unfortunately, human error is bound to happen; it cannot be predicted, nor could it be completely prevented. As it was mentioned in the Examples section earlier in the paper, one of the major database corruption incidents at JP Morgan Chase was caused by an operator executing the wrong script, and by doing so, moving the database index from the system to the user space. In multiple other instances, a database corruption problem was aggravated, because while the database operators were trying to fix the issue, they actually made it worse. The situation is even worse, when there is no track of what actions were performed "to fix" the problem, as these need to be communicated to the vendor support, in order for the vendor to understand the problem, and be able to resolve the issue.

5.2.2. Software/Application fault

It is relatively easy for an application which has write access to a database to cause a database corruption. For example, improper exception handling, or the lack of such, could result in non-sense (bad) data being written to the primary. The primary database has no way to know if the transaction that is applied to it is good or bad; if it has the correct SQL syntax, it will be accepted.

The database software itself can be the cause of database corruption. For example, if a new and not fully mature version of the database software is deployed, there is a risk that unknown defects are present in the software, which could result in database corruption on those systems. It is considered best practice to be a year behind with the

newest technology, so that the defects are discovered by the other early adopters, and fixed by the vendor.

Just like new versions, old versions which are out of support, can cause serious problems. If such a version is still used in the company, and a problem does happen, then the database operators would not be able to rely on vendor support's expertise, and it is likely that more time will be required to fix the problem.

5.2.3. Hardware fault

Database corruption may be caused by defects and faults on the server computer, especially the HDD (hard disk drive), disk controllers, the computer's main memory and the cache memory of RAID controllers. Sometimes hardware RAID controller can be slightly broken and cause random database corruption to occur.

5.2.4. Capacity/Volume Related Problems

Financial organizations like JP Morgan Chase deal with huge amounts of data every single day, every single hour. The amounts of the data are not constant, they are growing in size. This fact poses a serious challenge in front of database architecture and support groups. A quick glance over the problem data from the previous 3 years is enough to reveal that a significant portion of the outages and database problems reported were due to capacity and volume related problems. Fortunately, these problems have not caused prolonged outages, and were not hard to fix by adding extra hardware or other resources. However, those little outages add up, and eventually account for a healthy portion of the company's application outages due to database problems.

5.2.4.1 Database space

When the database fills up, it cannot be used. The applications and users that are trying to access it get error messages, the database support teams are informed, they fix the problem temporarily by adding resources, and then, sooner or later, the same scenario is repeated. Is that necessary? No. The database capacity should be automatically monitored, and upgrades should be planned ahead carefully, in order to avoid outages due to insufficient space in the database. Extreme conditions, like having a full database, only increase the chance of database corruption. Corruptions often occur in moments like the one described above, when the database capabilities are strained to the maximum, while at the same time more new incoming transactions demand even more space from the already full database.

5.2.4.2 Log space

For databases that are running in archive mode, the database space is not the only thing that could be brought to maximum. The database log space is also a potential problem. It has been a common problem in JP Morgan Chase. Several of the significant outages that we reviewed, mentioned log space issues as one of the causes. It is rarely a cause of database corruption; more often it is a symptom that something is not going right. When the log space is completely full, the database stops working, as there is no space to record the transactions that need to be executed.

5.2.4.3 Temp table space

Every database will become unusable, as soon as its temp table is full. This situation again puts the database in an extreme condition, and so a database corruption is likely to take place. As with the log space, temp table space filling up is often a symptom that there is something wrong going on, and an inspection is needed before things get worse.

5.3. Prevention of database corruption

Completely preventing database corruption is not possible. To achieve complete prevention, one would have to know all possible causes of database corruption, and how to deal with each one of them. As of today, that is not possible, and it probably will not be possible for a while. One thing that could be done, in order to achieve protection from database corruption, is to prevent the most common causes of database corruption. Below are described the strategies used for prevention of database corruption.

5.3.1. Monitoring

As it was mentioned several times up to this point, monitoring is a best practice, when it comes to database corruption. There are many monitoring solutions available on the market. Depending on the how critical the application is, the investment made for monitoring solutions will vary. For the most critical applications, investing in monitoring is highly recommended.

5.3.1.1. Capacity monitoring

Each database has limited resources. When there is a strain on the resources, the database often becomes unavailable for use. Monitoring the capacity could prevent that. Automating the monitoring could make it all fast and easy.

As it was mentioned above, database table space, log space, and temp table space are a few of the things that should be monitored for, as exceeding their limits could make the database unavailable and promote corruption. Other aspects that should be taken into consideration in the monitoring solution are the number of connections to the database, the current version of the database, and the change and maintenance procedures that have been performed on the database.

5.3.1.2. Performance monitoring

Performance issues can be one of the first symptoms of database corruption. When performance goes down, this may mean that the system resources are strained. Situations like that are very risky, because database corruption could easily and unpredictably occur in such an environment. Performance could go down even if the system resources are sufficient; this is almost always a sure symptom of database corruption, or some other inconsistency that has to be fixed before the system crashes. Performance monitoring should be automated, and running in the background all the time. Also, there should preferably be an automated procedure to be executed if a database operator cannot immediately attend to the issue.

5.3.1.3. Version control

There have been instances of database corruption in JP Morgan Chase which could have been prevented, had automated version control been in place, and had the required maintenance procedures been implemented. Version control monitoring does not require any significant investments, so we highly recommend its implementation and usage in the company. This type of monitoring could also be very useful for providing an overall picture of the company's current technology situation. Then it would be easier to determine if the recommended configuration patterns recommended in the bank have been implemented, and to what extent.

5.3.2 Database validation

Database validation involves checking the database files to ensure that the various data structures retain their integrity and internal consistency. The validation process should take into account different types of problems:

- **Corrupt data structures**: for example, if a database row spans more than one data page and the pointer that links the first data page to the second is damaged or missing, there is a corrupt data structure.
- **Misallocated data pages**: for example, a page can be used for transaction inventory, header information, data, blob pointers, or indices. If a page has been flagged as one type, but actually stores data of a different type, the database validation software should be able to detect the problem. However, if the validation software cannot recover from this type of problem, it will probably be necessary to restore from a backup.

A common way to validate the database is by using checksums. A checksum is a page-by-page analysis of data to verify its integrity. A bad checksum means that a database page has been randomly overwritten (for example, due to a system crash). However, the checksum method is not powerful enough, and will not protect against all problems. Because of this, most database vendors have developed validation software, which is much more powerful than checksums and usually protects from all known problems.

5.3.3 Discipline

A good way to protect against database corruption is discipline. By discipline, we mean being aware of and conforming to the established best practices, following the standard procedures, and performing everything in a structured and logical manner. When all actions are performed properly, and in order, while following established practices, the chance of database corruption, especially due to human error, is significantly reduced. Corruption can still take place, but as long as discipline was maintained, it is much easier to isolate the corruption and fix it, and to determine the root cause later. It is hardest, and at the same time most critical, to maintain discipline in times of failures and disasters, especially when the stress levels are high. Extensive training and practice is needed in order to achieve that.

5.3.4 Database operator training

Database operator training is one of the factors that contribute to discipline. When a new system is deployed, the operators should receive proper training, preferably directly from the vendor. When a problem needs to be solved by the database operators,

if not enough training was provided, the operators are more likely to make mistakes, thus aggravating the problem, rather than fixing it.

5.3.5 Simplicity

"Keep it simple", advises us one of JPMC's DBA's informally. The more complex the system is, the easier it is to break it, and the harder to fix it. In today's world, however, nothing is going in the direction towards simplicity; everything is getting ever more complex, including databases, applications, and the infrastructure around them. Consciously trying to avoid complexity and to maintain simplicity is surely going to pay off in the long run. Maintaining simplicity is not impossible for organizations of the size of JP Morgan Chase; other organizations with systems of similar size have maintained their databases simple, and have had relatively few issues with them.

5.3.6. Replication methods and database corruption

One of the most effective and common ways to prevent the expensive disasters and outages caused by database corruption and various other technical problems is to have reliable Disaster Recovery (DR) and High Availability (HA) solutions in place. DR refers to the preventative measures using redundant hardware, software, data centers and other facilities to ensure that a business can continue operations during a natural or man-made disaster and if not, to restore business operations as quickly as possible when the calamity has passed. HA refers to the virtually uninterrupted operation of a system during any given year. A system with 99.999% availability experiences only about five minutes of downtime. In contrast, a high availability system is defined as having 99.9% uptime, which translates into a few hours of planned or unplanned downtime per year.

DR and HA strategies include replication and backup/restore. Replication is the process of duplicating mission critical data from one highly available site to another. The replication process can be synchronous or asynchronous; duplicates are known as clones, point-in-time copies, or snapshots, depending on the type of copy being made.

In asynchronous replication, after data has been written to the primary storage site, new writes to that site can be accepted, without having to wait for the secondary (remote) storage site to also finish its writes. Asynchronous Replication does not have the latency impact that synchronous replication does, but has the disadvantage of incurring data loss, should the primary site fail before the data has been written to the secondary site.

In synchronous replication, each write to the primary disk and the secondary (remote) disk must be complete before the next write can begin. The advantage of this approach is that the two sets of data are always synchronized. The disadvantage is that if the distance between the two storage disks is substantial, the replication process can take a long time and slows down the application writing the data.[13]

Replication can also be classified as physical or logical. Physical replication is also referred to as disk mirroring. In physical replication, each physical write to disk is replicated on another disk at another site. Because the replication is a physical write to disk, it is not application dependent. This allows each node to run different applications under normal circumstances. Then, if a disaster occurs, an alternate node can take ownership of applications and data, provided the replicated data is current and

13 http://www.microsoft.com/windowsserversystem/storage/storgloss.mspx accessed Dec 2005

consistent.[14] The latter directly implies that physical replication/mirroring does not provide any protection against any kind of database corruption; whatever happens to the primary will eventually reach the secondary, no matter whether synchronous or asynchronous replication is used. Physical replication is typically synchronous.

Logical replication, on the other hand, is based on replicating data by repeating the sequence of transactions at the remote site. Logical replication can be configured to use synchronous or asynchronous writes. Logical replication can be implemented to reduce risk of duplicating human error. For example, if a database administrator erroneously removes a table from the database, a physical replication method will duplicate that error at the remote site as a raw write to disk. A logical replication method can be implemented to only replicate database transactions, not database commands, so such errors would not be replicated at the remote site. This also means that administrative tasks, such as adding or removing database tables, have to be repeated at each site.[15] Logical replication, because of its nature, would also protect against hardware-induced database corruptions – something physical replication would not do.

The conclusion here should be that, in order to have a reliable protection from database corruption, logical replication must be involved in the DR/HA solution. If no data loss is desired, then replication should be done in synchronous mode. If speed or performance is the priority, then replication should be done in asynchronous mode.

14 Disaster Tolerant Architecture Guidelines, http://docs.hp.com/en/B7660-90014/ch01s04.html accessed Dec 2005

15 Disaster Tolerant Architecture Guidelines, http://docs.hp.com/en/B7660-90014/ch01s04.html accessed Dec 2005

5.3.7. Vendor support level

Most vendors offer different levels of support for their products. The highest level of support, also commonly referred to as "gold-level support", provides a dedicated team, which the database administrators can speak to directly when needed. Without gold support, the 800 number is the way to reach the support personnel, which may or may not be able to provide the quality and urgency of the help needed. Investing in "gold-level support" should definitely be considered for the mission-critical systems, but it should not be entirely relied upon; it would be much better to be able to rely upon in-house expertise and the established prevention and configuration practices.

5.4. Dealing with database corruption

There are many hardware and software solutions that could be used to deal with the problem of database corruption, each having relative advantages and disadvantages. There is no best solution that could cover all problem scenarios; rather, the best solution will vary, depending of the specifics of the application/system. The goals associated with dealing with database corruption, however, are the same everywhere: faster failover to a good copy of the database, minimal data loss associated with the failover process, and corruption isolation. The latter is probably the most important goal, because if the database corruption is spread through the backups, they are practically unusable copies of the database. Then the only way to bring back the application to life is to restore from yesterday's tape. There is a risk that this tape contains the corruption as well, as corruptions are sometimes present in a system for an extended period of time, but they are not noticed since no monitoring or usage has been done on this particular set of data.

Days could go by before a good copy is found to restore from. Meanwhile, all the transactions that have occurred since then would have to be re-applied, so it is critical to backup the transaction logs, as well as the data itself. The conclusion here should be that it is quite difficult to deal with database corruption. There are many solutions one could invest in, but not all of them would be appropriate or affordable, so careful consideration should be given to the decision of which solution to use.

Dealing with database corruption is made harder by the fact that database corruption happens quite rarely, so there is limited knowledge base on it. As a result, database corruption is not always taken into consideration when designing HA/DR solutions, and when a database corruption occurs, it could be quite devastating and hard to deal with.

5.4.1. Human intervention

The life-cycle of database corruption usually begins with the following sequence of events: user experiences problems with the application, they call the helpdesk, and the helpdesk operators determine that the problem is database corruption, and try to fix it. Human judgment is involved in two critical stages – classifying the problem as database corruption, and deciding how to deal with it. The sad truth is that humans, no matter how experienced, do make errors in their judgments. Human intervention could be the best and the worst that could be done in case of database corruption. An automated process may not accomplish what an experienced database expert could, but at the same time, it might not hurt the system as much as an inexperienced/careless/distracted database operator. The conclusion: human intervention could resolve or aggravate the issue. It is an irreplaceable way of dealing with database corruption and all other kinds of problems,

but at least actions should be recorded during the recovery process, so that if it comes to vendor support, it is known for sure what actions were taken.

5.4.2. Vendor support

There are varying levels of vendor support, and which one is used depends on the criticality of the system, and the resources available. The more the company pays for vendor support, the quicker it is usually to resolve issues like database corruption. At JP Morgan Chase turning to the vendor for support is a common practice, per our observations. Providing the database administrators with a dedicated support team would shorten the time to recover. The investment into gold-level support could go for operator training instead, and this would be much more beneficial for the company overall, because it would make turning to the vendors for help obsolete, prevention of database problems better, and it is very likely to reduce the number one cause of database corruption, namely human error.

5.4.3. Snapshots / Checkpoints

Snapshots, which are also called checkpoints, have been mentioned on several occasions so far as a good way to deal with database corruption. A snapshot is a point-in-time image of the database. They are very fast and inexpensive to take, as they only require the primary database to freeze for a few seconds, while the snapshot is being taken. Snapshots are small in size; instead of copying the entire dataset like a mirror or clone, a snapshot is a freshly created and separately stored index of pointers that show where primary data is stored at a certain point in time.[16]

16 www.fcw.com/fcw/articles/2004/0809/feat-backup1-08-09-04.asp accessed Dec 2005

5.4.4. HA/DR solutions vulnerable to database corruption

There are some excellent HA/DR solutions in place in the bank, which account for virtually everything but database corruption. Below we are going to point out those solutions that should not be used alone, as they will not protect against a database corruption disaster.

5.4.4.1. Physical block replication / Mirroring

Physical block replication is a technology vulnerable to database corruption. This directly implies that other technologies based on physical replication, such as mirroring, are also vulnerable to database corruption. It does not matter what kind of corruption occurred on the primary; it will be instantly transferred.

5.4.4.2. Clusters

According to Microsoft, "Clustering in conjunction with a failsafe storage device system represents a true fail-over solution *for everything but data corruption*. It is limited by expense of hardware, difficulty of configuration, shared media, controller failure potential, and the fact that it offers no protection against database corruption."[17] The conclusion is that some other mechanism, like snapshots, should be used in conjunction with clustering in order for the databases to be protected against corruption.

5.4.5. Featured solutions

In the following sections we are going to present some solutions that could improve both protection and mitigation strategies against database corruption. They are

17 www.microsoft.com/technet/ prodtechnol/sscomm/reskit/ss3fop.mspx accessed Dec 2005

just a small subset of what is available on the market today, but they are fairly representative of the options for HA/DR that exist, and that provide additional features targeted specifically at database corruption.

5.4.5.1. Sybase Mirror Activator

Introduction to the product:

Sybase Mirror Activator is a business continuity solution that lowers overall total cost of ownership (TCO) by reducing failover time and network bandwidth requirements and making the usually idle standby database available for reporting and decision support. [18] According to Raj Nathan, senior vice president, Information Technology Solutions Group, Sybase, "The result is improved production performance caused by moving query workload to the standby database. With Sybase's Mirror Activator, we enable the mirrored data to be fully utilized for everyday reporting or maintenance, and the warm standby database significantly shortens recovery times."

18 http://www.continuitycentral.com/news01315.htm accessed Dec 2005

System architecture:

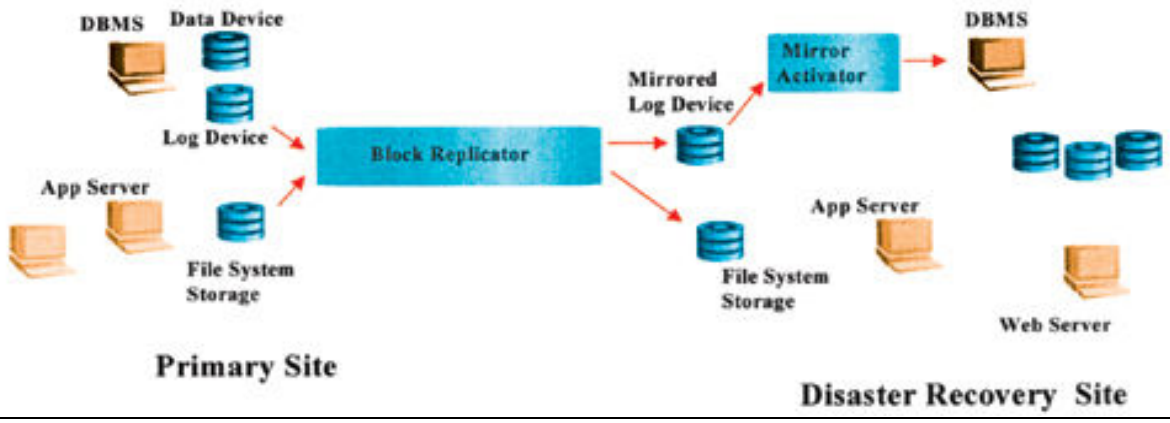

Figure 3: Sybase Mirror Activator

[19]

Advantages:

* Increased data availability. With the warm standby database, a secondary site is always available and failover of applications can be accomplished in seconds instead of hours.

* Reduced network bandwidth costs. Organizations can lower the required bandwidth by up to 50 percent, dramatically reducing TCO.

* Improved return on assets. The standby database is always online and available for read-only client applications like reporting and analysis, improving the ROA for disaster recovery hardware.

In the context of database corruption:

Mirror Activator relies on its Replication Agent component for corruption protection. This solution will protect against physical corruption, because transactional

[19] Sybase.com accessed Nov 2005

replication is involved, but it cannot do much about logical corruption. Again, the use of snapshots is advised if this solution is used.

Disadvantages:

Mirror Activator is very expensive, and it works only for Sybase at the time of writing this paper. Also, it is a relatively new and still not fully mature product, so it is not completely known whether the product really delivers the features it is said to deliver.

Conclusion:

We would recommend that JP Morgan Chase perform and carefully evaluate the Proof of Concept experiment in order to determine if the product could really prevent outages like the M3 outage from March 2004 (described in the "Examples" section). If a better price is negotiated, and the Proof of Concept experiment turns out to be successful, in addition to Sybase Mirror Activator supporting other databases besides Sybase, and the reviews of the product from the early adopters are positive, only then it would be a smart step to invest in this product.

5.4.5.2. Oracle Maximum Availability Architecture

Introduction to the product:

The Oracle Maximum Availability Architecture (MAA) is not a single product; rather it is a combination of Oracle products, which together have proved to provide one of the most resilient architectures available today. MAA includes Oracle10g database, Oracle Real Application Cluster (RAC), Oracle DataGuard, and Oracle FlashBack (which comes integrated with Oracle10g at no extra cost). MAA is a mature, tested architecture which has proven to be effective. There are many whitepapers and presentations written on it. There are numerous companies which have successfully implemented it and have

recommended it to other companies. MAA is considered a best practice for providing high availability for mission-critical systems.

System architecture:

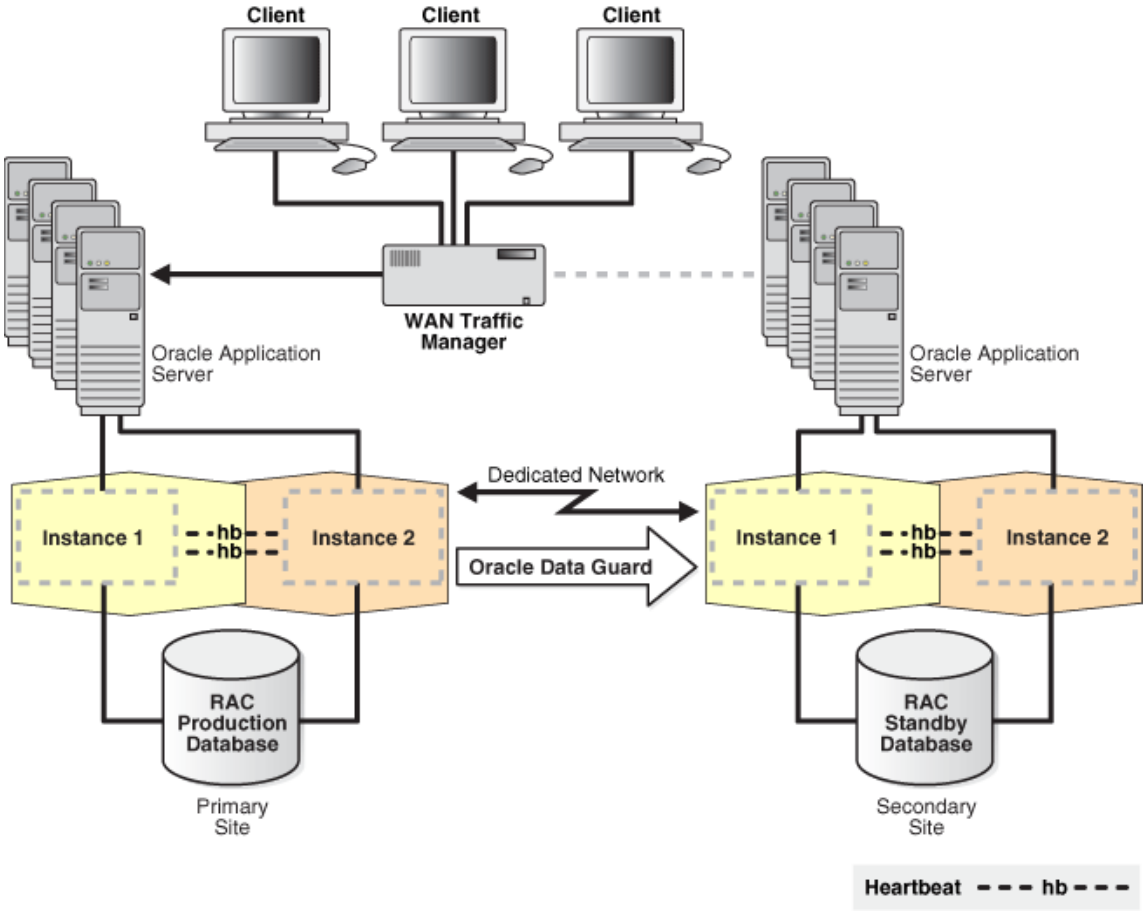

Figure 4: Oracle 10G RAC

[20]

Advantages:

- Highly integrated and tested architecture
- A lot of documentation/whitepapers/case studies available
- Flashback technology effectively addresses human errors

20 http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm accessed Dec 2005

In the context of database corruption:

There are three features of the MAA that are specifically targeted at protecting against and dealing with database corruption. Those products are:

- Flashback database
- Hardware Assisted Resilient Data (H.A.R.D.)
- Recovery manager (RMAN)

RMAN has been officially recommended by the Engineering Board at JP Morgan Chase as a good solution not only for database corruption protection, but also for providing disaster recovery and high availability is general. Its real power, when database corruption is concerned, is that all data blocks can be analyzed for corruption during backup and restore, to prevent propagation of corrupt data through the backups. RMAN also has capability to recover corrupted database blocks while the datafile remains online.

Disadvantages:

The cost of implementing this architecture could be significant. Implementing MAA should only be considered for mission critical applications, which have the responsibility to protect critical data. Another potential disadvantage is the complexity of the MAA. Database operator training would have to be provided in order to make sure all the components of MAA are fully utilized and correctly configured. Last but not least, MAA works only with Oracle.

Conclusion:

Highly recommended for mission-critical Oracle systems and mission-critical data. Training should be provided. Configuration should be made extremely carefully.

5.4.5.3. GoldenGate Transaction Data Management Software

Introduction to the product:

GoldenGate's Transactional Data Management (TDM) product enables their customers to effectively maximize the performance, accessibility, and availability of the transactions that drive their mission-critical business processes. TDM claims to provide guaranteed real-time capture, routing, transformation, delivery, and verification of data transactions across heterogeneous environments. The transactions are moved with sub-second latency, making the system a real-time system. It is able to handle thousands of transactions per second, making it a high-performance system. The architecture of the system is extensible, so it is easy to adjust the product to the specific needs of the customer. The system is based on transactional replication, so transaction integrity is guaranteed. Last but not least, it is possible to move transactions across different databases and platforms – a feature rarely seen with other solutions.

GoldenGate has been providing Transactional Data Management solutions for more than 10 years now.[21] GoldenGate has a long list of clients whose business is similar to JP Morgan's business. The retention rate of GoldenGate's clients is in the high nineties, so this is a trust-worthy sign of the product quality.

[21] http://www.goldengate.com/resources/ accessed Dec 2005

System architecture:

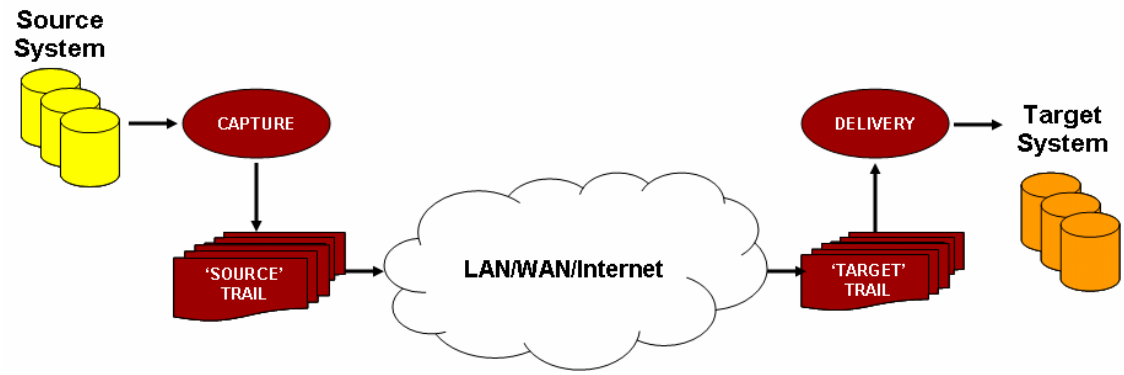

Figure 5: GoldenGate

22

[22] GoldenGate presentation for JP Morgan Chase, July 13 2005

Advantages:

Using GoldenGate TDM for data replication offers significant advantages over the traditional replication approaches:

Traditional Replication	**TDM**
Like-to-like databases and platforms	**Cross-platform** databases and platforms
One-to-one topology with all-or-nothing data replication	**Many-to-many topology** with selective and optimized data movement
May not guarantee transaction integrity	**Guaranteed transaction integrity**
Data corruption propagated to target	Data **corruption is isolated** at source
No rollback or point-in-time recovery	Selective and **dynamic rollback** with advanced point-in-time recovery
Target system unavailable for data processing	Both source and target systems are **available for uninterrupted business support**
No data transformation capability	Data **transformation** and enrichment
Geographic distance limitation between data source and target	**No distance limitations**

Figure 6: Comparison Chart
[23]

In the context of database corruption:

TDM claims to be able to do what many other systems fail to provide: isolation of database corruption at the source, which means TDM would prevent problems like the M3 problem described previously from happening. The selective and dynamic rollback with advanced point-in-time recovery would mitigate the effect of human errors, and would shorten the recovery time in case of database corruption issues. Also, since the latency involved is low, data loss is brought to the minimum. TDM provides a warm standby with corruption isolation at the primary, which is an established best practice.

[23] GoldenGate presentation for JP Morgan Chase, July 13 2005

All of the above features are provided at a lower cost by GoldenGate, compared to other component providers. TDM could be easily integrated with third-party monitoring products, which is a significant advantage since monitoring is critical for protection against database corruption. GoldenGate's VeriData could be used in addition to TDM to identify and report data discrepancies between the primary and the backups, which is a way to detect database corruption. VeriData provides an easy way to provide quality control of the database and failover systems, thus potentially mitigating the effects of database corruption. VeriData will proactively discover discrepancies before they become problems, thus providing preventing database corruption before it causes any problems or outages. Dynamic rollback and selective recovery capabilities support the best practice of maintaining point-in-time images of the data for fast recovery and mitigation in case of database corruption.

Also, GoldenGate TDM does not have the distance limitations products like MirrorActivator have, which is another advantage of this product. This comes in addition to all systems being synchronized all the time – something traditional transaction replication cannot guarantee. Last but not least, the use of GoldenGate TDM will facilitate future migrations to more stable systems (Sybase to Oracle, for example) without having to purchase new technology to provide high availability and data protection.

Disadvantage:

The product and the company are relatively new, so we recommend that JP Morgan follows its usual best practice to be a step behind with the newest technology, and to wait until more white papers are published on it, and more companies adopt the

technology and discover any hidden bugs that might exist. Some people might view the lack of physical block replication as part of this solution as a disadvantage, but we believe that if the product is indeed able to keep remote databases synchronized without latency, and failover really takes minutes or seconds, it should not be regarded as a disadvantage.

Conclusion:

Highly recommended.

6. Business Continuity:

6.1 Industry Trends

EnvoyWorldWide conducted a survey to identify business continuity and disaster recovery practices and trends. Some key findings of the survey are:

Top trends in 2005:

1) The management of the business continuity process has evolved into a collaborative initiative within organizations rather than a siloed activity.
2) Regulations and customer requirements drive companies to complete, test and distribute their business continuity plans.
    a) Over seventy five percent of the companies surveyed cite that federal, state or industry regulations directly affect business continuity initiatives.
    b) Sixty six percent of the respondents observe an increased interest in their business continuity plans from their respective customers while sixty eight percent have seen BCP requirements in RFPs and RFIs received.
3) The number of companies that migrate manual calling solutions to automated notification solutions has dramatically increased. [24]

These trends are in line with our findings at JP Morgan. The fact that the groups under John Storm are collaborating is an example of the first trend. This is a good way to approach this problem as many times in large organizations, like JP Morgan, some groups will see things one way and others will see it a different way; having more than

[24] http://www.envoyworldwide.com/news/releases/061305.shtml

one perspective helps find hidden best practices because if just one group is looking to solve something they will have a much harder time looking beyond their current frames of thinking. The second trend rings true at JP Morgan as a company of this size will not be able to fly below the radar of the government and will not be able to ignore any concerns customers may have. That being said, they must satisfy both of them.

The movement to migrate from manual solutions to automated ones has become important at JP Morgan too. In most of our interviews we saw that most people felt that the problems occurred when DBA's had to try to fix a problem and did not follow proper procedure or caused more problems. The ideal solution is one that is completely automated, but as none seem available yet, the requirement is to automate everything else, specifically the monitoring and notification.

6.2. Cost of Downtime and Data Loss

Obviously, in a major organization such as J.P. Morgan Chase a database outage is incredibly costly, but exactly how much does it lose when an outage occurs? This proved to be a rather difficult question to answer. The usual response is that the greatest loss is "opportunity loss" and it is "incalculable." While this is certainly arguable, it is not definite. The following sections will show various ways to look at the problem at hand.

6.2.1. System Criticality

As clearly stated, JP Morgan has some of the most critical systems in the world, some, in fact, can affect the world financial markets. Many of these systems cannot afford to have a minute of downtime, or else the impact, financial or otherwise, will be enormous. That being stated, the integrity of their infrastructure is of great concern. But, which applications need to be up and running constantly and which can spare some

downtime? At the bank, there is a list of the top 45 systems that are considered critical to the business and if these were to go down the bank would be in a dangerous situation. Beyond those 45, each team must decide the criticality of their system, considering what is most important to that system.

The top 45 cannot afford any amounts of downtime or data loss. Each system must consider which is most important to them. For instance, pricing systems that refresh themselves with market data are quite sensitive to uptime. If the system is down and is not updating market data, the bank could be losing money. In the same light, frontline trading systems must have little downtime because that would mean traders could not do business, which for obvious reasons, cannot happen. For these systems, it would be recommended to have a failover system with low latency, such as Sybase Replication or Mirror Activator.

On the other hand, there are systems that are not as concerned with uptime, but more with data loss. For instance, STS is a system that does not require immediate failover, but cannot afford to lose any data. It could lose an hour and the bank would be fine, but to if it were to lose 72-hours like the M3 outage, the bank would be, as Tom McLean put it, "out of business." For these types of systems, it would be ideal to have some type of automated snapshot technology, such as Oracle Flashback, that would allow for a quick restore to a point when the data was complete rather than restore from tape. The advantage of this is the fact that it would allow maximum data recovery with a moderately quick restore.

For the applications that do not have either one of these elements as primary concerns, they must decide based on the nature of the system. For instance, SAMPRAS

could be considered one of these systems that it is not critical in terms of data loss or latency, but obviously it is best to avoid long outages. The nature of this system is that it has a long batch that calculates aggregate risk numbers and if they get some bad data, it could cause it to freeze or cause the aggregate numbers to be incorrect. To fix this problem, the team would need to restart and that takes about 12 hours to do. Considering this, they would want to invest in a snapshot option that would allow them to flashback to a set of complete data without having to lose 12 hours. As you can see there is a range of problems with a range of solutions. Some solutions will fit more than one business's needs. Some will need to be custom fitted for the specific requirements of an organization, but all solutions should aim to meet all requirements.

6.2.2. Reputation Loss

An organization's reputation is obviously something that is very important to the success of said organization, especially in times where it seems corporate scandals like the ones with Enron and Worldcom seem more and more prevalent. Things can be done to try to create positive brand equity, such as strong marketing and public relations, but nothing can really be done when a company's name becomes tainted. Blows to an organization's reputation do not come with just major scandals like these. For example, when users cannot access systems, the company runs a high risk of depreciating the value of their name and potentially losing customers. Also, the reputation will become exponentially lower the longer the outage or the more frequently they occur.

The opinion of the organization could be worsened by competitors in the same industry who are willing to capitalize on the company's weakened brand equity. Losses to reputation and trust are not easily gained back and could potentially bring an

organization down. Obviously, no one can tell how much is actually lost when brand equity is lowered and that is perhaps why it makes this type of loss the most damaging in some people's eyes.

6.2.3. Financial Loss

According to Karen Hengerer, historically, only a handful of the 600+ application management teams had produced actual numbers after an outage occurs. This interested her because years ago for her Masters thesis, she had discovered the Gartner Group came up with a number of about $150,000 in opportunity loss per hour of downtime. Their algorithm calculated an actual dollar figure for outages on 50 of the most critical applications, mostly trading applications. They based their algorithm on 2004 IB revenue for all three regions, which was about $13.5 billion. The algorithm has been expanded by the JPMC team, from 50 of the most critical applications to the top 200. The applications considered are the largest revenue-producing applications or if it affects one of the largest revenue-producing applications. Below are general statistics of the algorithm, including customers

impacted.

2004 IB Revenue	$13,500,000,000.00		
	50 Apps	200 Apps	Difference (50-200)
Financial Impact Per Business Per Hour (100-57-30-13)			
FB = All	$6,750,000.00	$6,750,000.00	-
FB^1 = NA	$3,847,500.00	$3,847,500.00	-
FB^2 = EMEA	$2,025,000.00	$2,025,000.00	-
FB^3 = AP	$877,500.00	$877,500.00	-
Financial Impact of a Single App Per Hour (based on 50)			
FA= All	$135,000.00	$33,750.00	$101,250.00
FA^1 = NA	$76,950.00	$19,237.50	$57,712.50
FA^2 = EMEA	$40,500.00	$10,125.00	$30,375.00
FA^3 = AP	$17,550.00	$4,387.50	$13,162.50
Customers Impacted by Loss of Entire Business			
CB = All	8000	8000	-
CB^1 = NA	4000	4000	-
CB^2 = EMEA	2000	2000	-
CB^3 = AP	2000	2000	-
Customers Impacted by a Single App (based on 50)			
CA = All	160	40	120
CA^1 = NA	80	20	60
CA^2 = EMEA	40	10	30
CA^3 = AP	40	10	30

Figure 7: Algorithm Results[25]

Using this algorithm, the M3 outage, which was three days long, had a financial impact of between $1,385,100.00 and $5,540,400.00. The algorithm not only gives concrete numbers to work with to calculate loss, but it gives a guideline for teams affected by an outage to discuss how much they potentially lost, allowing them to claim a better ballpark figure. This is obviously a valuable tool as it allows the organization to better estimate what losses occur with outages, permitting it to better plan financially for ways to mitigate losses caused by downtime.

[25] Internal document: Algorithms for Oppurtunity Loss_05_24_05.xls

6.2.4. Loss of productivity

Obviously when a system goes down, there are financial losses and to reputation, but there is another cost to be analyzed here and that is the loss of productivity. Outages force employees to stop working simply because their work or system is unavailable. This costs the company the wages paid to those employees while they are not working. Simultaneously, the company must pay the team that is called in to fix the outage. Below is an equation that shows the amount of loss of productivity, which is based on the number of employees sent to fix the problem (On average, a team consists of 3-to-5 employees.) and the employees who are unable to do work (between 25-to-500 employees, depending on the type of system). Other factors are avg. hourly pay per JP Morgan employee (which is $100-$125) and the duration of the outage. The averages were provided by Karen Hengerer.

$$\text{Loss of Productivity}$$

$$Z = (X_1 + X_2)*Y*t$$

- Z = Productivity Loss
- X_1 = # of employees to fix outage
- X_2 = # of unutilized employees
- Y = avg. hourly pay per employee
- t = duration of outage

Given this equation, let's look at some possibilities for the financial amount lost during an outage because of unutilized employees. Say an outage occurs and systems are out for eight hours (t = 8). A team of three members is assembled to fix the issue ($X_1 = 3$). The number of employees who are being paid but cannot work depends on what system they work with. For example, if a system that deals with productivity tools goes down, such as Sametime, email or Blackberry service, about 100-500, on average will be

affected and unable to do work. If development tools went down and drive connectivity were to go down, they would each cause 25-50 people to become unproductive on average. These numbers are just averages, but obviously, they can be higher or lower. That being stated, let's say that email went down and 150 people were affected (X_2 = 150) and the average salary of these people and the fix team is $115/hr (Y=$115). Given these numbers:

$$Z = (X_1 + X_2)*Y*t = ((3 + 150) * \$115 * 8 \text{ hr.}) = \$140,760 = \$17,595 / \text{hr}$$

For a single work day of outage, it would cost JP Morgan $140,760 in this scenario. The equation seems simple, but it is obviously a very complex problem when an outage occurs and it reinforces how much outages can cost the company. Naturally, one would want to fix an outage as soon as possible or, as stated throughout this document, set up a system where there would be no downtime and/or no data loss. This equation, along with the algorithm explained earlier, helps put costs into perspective.

6.3. Cost of Investing in DB corruption prevention/mitigation

It has been made clear that business continuity is a necessity in business today, but the question remains "How much is this going to cost?" Obviously, no organization wants to suffer from database corruption, but similarly, no organization wants to break the bank to protect itself. So, it is better to ask, "How critical are the things we need to protect?", "How much will an outage cost?" and "How much are we willing to pay to avoid that cost?" The first two of these questions have been answered. The last one will be answered here. Actual prices for products and services from various companies were

either unavailable or unquotable, so no actual figures will be used. We will, however, discuss a process how to determine what your business needs to weigh against price.

Considering how much can be lost during outages, price really comes second to the needs of your business continuity plan. The differences in prices of different packages to purchase are minor, especially considering how the wrong solution could be largely detrimental to the infrastructure of the bank. For instance, if your system really needs seconds to failover and what you purchase takes minutes to failover, the time lost in availability could cost you more than the difference you paid for the solution. This could be even made even worse when the solution you purchased works in less than ideal conditions, creating even more losses.

This scenario is obvious, but it is something that is necessary to be explained, especially in such tight financial times where downsizing is not uncommon. The key is to save the most money in the long run and not at the point of purchase, although saving money at that point on a good product would certainly be well received.

With the point of prioritizing your needs stated, we must now consider cost. The combination of loss of reputation, opportunity and productivity equal your total loss. This figure must be weighed, in conjunction with your business requirements, specifically for data loss and downtime, against the cost of proposed solutions. Considering what we know about database corruption and its costs, we will take a look at various solutions and what they can provide in terms or availability and data loss protection.

For Sybase:

	High Data Loss Protection	Low Data Loss Protection
High Availability	Mirror Activator	Sybase Replication
Low Availability	Snapshots	Tape Restore

Figure 8: Sybase Matrix

For Oracle:

	High Data Loss Protection	Low Data Loss Protection
High Availability	Recovery Manager	Oracle 10G RAC
Low Availability	Flashback	Tape Restore

Figure 9: Oracle Matrix

Obviously these are just a few of the products we looked at, but it must be reiterated that your business solution is dependant on your specific needs and only you can classify those to figure out the solution to your high availability and data loss concerns.

7. Conclusion

Database corruption is unpredictable and unavoidable. All database systems are vulnerable all the time, as are all electronic systems. There is no way to completely prevent database corruption from happening; a database can become corrupted at any time for no discernible reason. There are different ways to mitigate the negative effect of

database corruption, which are based on frequently backing up the primary, protecting the backup copies of the database from being corrupted, and quickly failing over to those copies in case of corruption on the primary, while minimizing the data loss. The quality and effectiveness of the different types of solutions varies, and so does their cost of implementation. Unfortunately, the most effective solutions can be viewed as an "expensive insurance" because of the high total cost of ownership (TCO) and low return on investment (ROI) that is typically associated with them. Cost-benefit analysis shows that for the most critical systems, the costs associated with those solutions are justifiable.

7.1. Mirror Activator

Sybase Mirror Activator claims to be the only product that has very low latency (as low as 9 seconds) and no data loss. Considering how new this product is, it is difficult to confirm or disconfirm these claims. There will be, however, a proof-of-concept trial for JP Morgan this month and the results should be revealed January 2006. From all interviews conducted, most experts did not fully recognize database corruption as an serious issue and saw database outages as inevitable. Considering the data, they naturally did not see Mirror Activator as a valuable product.

One Sybase DBA who chimed in similarly had something more to add. He said that the main difference between what is currently installed and what Mirror Activator provides is faster replication for applications handling high load data transactions, like inserting or deleting many rows at once. For those applications, he thought Mirror Activator would be valuable. He also added that Mirror Activator would be useful in the case of a database being destroyed and the bank needing to failover quickly to the DR site, but cannot mitigate the effects of someone running bad statements, such as deleting

something that should not have been. In those cases, he felt that the only thing to fix the problem was to recover from the latest transaction dumps.

Despite Mirror Activator obviously having value, it cannot be advised for JP Morgan to adopt Mirror Activator as of yet. The company has a history of being rather conservative when adopting new technology, and for good reason. Considering that JP Morgan is a crucial cornerstone in the world financial markets, they cannot afford to adopt brand new technology that may or may not work flawlessly. Any new product of this nature has bugs and glitches in it. It may be a small one that can be fixed in an hour or a large one that may take 24 hours, but obviously, JP Morgan cannot afford any lost time in their most critical applications. Once Mirror Activator matures to a product with proven near perfect consistency, it is advised that then and only then should JP Morgan adopt. Also, at that point, the cost of this product should have decreased significantly. On the other hand, time is of the essence. If there are critical applications that cannot spare any more time without a strong safety net similar to Mirror Activator, one could consider Golden Gate.

8. Future directions

There were a few issues we ran into during our time here and there are a few possible solutions that could possibly improve the processes at JP Morgan. Those suggestions are:

8.1. Better problem tracking system

A major issue we had when first arriving at JP Morgan was trying to find problem data. We spoke to many people about relevant database corruptions and we found none until a few weeks into the program. A solution to this problem would be a better problem tracking system, where you could find problem data by certain criteria such as root cause. This would help a group that's looking for trends or solutions to problems.

8.2. Maintenance procedures

Another thing that could be further analyzed would be the maintenance procedures at the bank. As M3 was a problem that could have been avoided if they had updated their version of Sybase, which was 4 years old, it is important to maintain the systems used by the bank, ideally automating the processes as much as possible so that no human errors can be made by doing something wrong or not doing something.

8.3. Detailed overview of infrastructure, configuration, and best practices in place

Along with these suggestions, detailed mapping of configurations and infrastructures of systems would be very useful so that processes can be looked at and so that best practices can be found too. With this, known best practices should be logged

somewhere in great detail so other groups can see what has already been discovered, thus saving time and resources by not having to relearn any lessons or rediscover best practices.

8.4. Implement recommended patterns

Once these detailed maps have been made, there could be a movement to move to recommended patterns suggested by individuals such as Felix Bodmer. The configurations he would like to see implemented are ideal for being most resilient. For more on these configurations see the internal document we looked at: Definition of Standard Configurations for Databases v1[1].5.3.doc.

8.5. Examine reducing complexity

While looking at all the systems, another thing to keep in mind is the complexity of a system. Every expert we talked with said the phrase “Keep it simple.” If the systems could become simpler without losing any functionality, that would be ideal. It is known internally that the more complex the system, the more likely it will fail and the harder it will be to fix.

8.6. Examine critical application upstream dependencies

Another thing to consider is looking at how the systems are related. There were many outages we ran into that could have been avoided if the critical system wasn’t dependant on unreliable systems like LotusNotes. These configurations are obviously not designed well since the most important applications should be more upstream, thus having fewer dependencies.

8.7. Design an automated disaster recovery procedure

An automated disaster recovery procedure would be obviously the best thing to mitigate the effects of database corruptions. As clearly displayed throughout this document, human error is a major contributor to the headaches that the IB is dealing with. To avoid these problems it would be best to automate the process to recover from any disasters, so that no DBA's put their hands into a problem and come out with even more problems, rendering systems unavailable for days and the bank losing $100,000's in opportunities missed or lost productivity.

9. Epilogue

Excuse the cliché but, working over the past seven weeks has been life-changing. It cannot be explained through words what it is like to do a project of this caliber with a company like JP Morgan. It can, however, be expressed through words how appreciative we are to the people throughout this organization. The time they have spent with us and for the knowledge they have provided can neither be replaced nor repaid, but all we can do is say, "Thank you." Without them, there would be no project.

Our project allowed us to be in a situation where we did not know all the details and answers. Our project created a situation that we had to move out from our personal comfort levels and into unfamiliar depths. Our project created a situation that was "real-world." Perhaps, it was just a glimpse of what it is like, but that brief glimpse allowed us to see something new outside of the world we know, and perhaps something new inside of ourselves. We can say that the time spent in New York has been invaluable and we will draw life lessons from it for a long time to come. Once again, we thank our sponsors for the opportunity and privilege to work within such an esteemed organization.

10. Bibliography

- http://ibm.com …Introduction to problem determination (accessed Oct 20, 2005)
- http://www.sybase.com … BCPDRv1.22May02.pdf (accessed Nov 3 2005)
- http://www.gartner.com … laws_influence_business_cont_128123.pdf (accessed Dec 5 2005)
- http://www.dmreview.com/resources/glossary.cfm?keywordId=ALL (accessed Dec 8 2005)
- http://jpmc-intranet.bankone.net/corphistory/timeline.asp (accessed Dec 9 2005)
- http://www.envoyworldwide.com/news/releases/061305.shtml (accessed Dec 9 2005)
- Internal Document: 01_Resiliency_Story-Summary_Jan2004_John_Storm.pdf (accessed Nov 30 2005)
- Internal Document: Roadmap to 3 9s.ppt (accessed Dec 7 2005)
- Internal Document: Concorde Archive Database Incident Report – 7/11/01 (accessed Nov 29 2005)
- Internal Document: TM outage.ppt (accessed Nov 29 2005)
- Internal Document: Operate – Trevor Outage (accessed Nov 29 2005)
- Internal Document: Algorithms for Oppurtunity Loss_05_24_05.xls (accessed Nov 20 2005)
- Internal Website: Appquest - M3 (accessed Oct 27 2005)
- http://www.ibexpert.info/documentation/Database/DatabaseCorruption (accessed Nov 2005)
- http://www.fcw.com/fcw/articles/2004/0809/feat-backup1-08-09-04.asp accessed Dec 2005
- GoldenGate presentation for JP Morgan Chase, July 13 2005
- http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm accessed Dec 2005

- http://www.microsoft.com/technet/ prodtechnol/sscomm/reskit/ss3fop.mspx accessed Dec 2005
- http://www.continuitycentral.com/news01315.htm accessed Dec 2005
- Sybase.com accessed Nov 2005
- http://www.goldengate.com/resources/ accessed Dec 2004
- Disaster Tolerant Architecture Guidelines, http://docs.hp.com/en/B7660-90014/ch01s04.html accessed Dec 2005