

UNIVERSIDADE LUTERANA DO BRASIL

ULBRA – *CAMPUS* GUAÍBA

CURSO DE SISTEMAS DE INFORMAÇÃO



***ORACLE RAC 10G* SOBRE SISTEMAS
VIRTUALIZADOS E *ISCSI*
TRABALHO DE CONCLUSÃO DE CURSO I**

RODRIGO DA ROCHA PACZEK

André Peres
Orientador

Guaíba, Junho de 2008.

DADOS DE IDENTIFICAÇÃO

Acadêmico: Rodrigo da Rocha Paczek

E-mail: rodrigobrar2003@gmail.com

Professor Orientador: André Peres

E-mail: andre.peres@ulbra.br

Título do Projeto: *Oracle RAC 10g* sobre sistemas virtualizados e *iSCSI*

Período de realização: 15/03 á 30/06/2008

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Delimitação do Tema.....	6
2 SGDB ORACLE DATABASE 10G.....	7
2.1 SGDB Oracle Database 10g.....	7
2.2 Tipos de bases Oracle Database 10g.....	13
2.3 Desempenho e tolerância a falhas em bases Oracle Database 10g.....	14
2.4 Bases Oracle Database 10g Stand-alone	18
2.5 Bases Oracle RAC 10g.....	19
2.6 Licenciamento.....	25
3 AMBIENTES DE ARMAZENAMENTO DE DADOS PARA BASES ORACLE RAC 10G.....	26
3.1 Storage	27
3.2 Ambiente Virtual	28
3.3 iSCSI	29
4 CENÁRIO DE TESTES	31
4.1 Requisitos de Hardware.....	34
<i>4.1.1 Virtualização</i>	<i>34</i>
<i>4.1.2 iSCSI</i>	<i>35</i>
4.2 Requisitos de Software	36
<i>4.2.1 Virtualização</i>	<i>36</i>
<i>4.2.2 iSCSI</i>	<i>36</i>
5 CONJUNTO DE TESTES PROPOSTOS	36
6 METODOLOGIA	37
7 CONCLUSÕES	41
8 REFERÊNCIAS BIBLIOGRÁFICAS	41

1 INTRODUÇÃO

O tema abordado nesta pesquisa diz respeito á possibilidade de implementação da tecnologia de clusterização de banco de dados *Oracle* através da *feature RAC (Real Application Cluster)* inserida na versão *10g* do *SGBD*, avaliando sua viabilidade e desempenho sobre sistemas virtualizados, bem como sua utilização sobre a tecnologia *iSCSI (Small Computer System Interface)*.

São utilizados neste projeto, *hardwares* e *softwares* disponibilizados pelo laboratório de informática da Universidade Luterana do Brasil, campus Guaíba, com o intuito de implementar o projeto, bem como realizar testes de desempenho entre as duas tecnologias abordadas – *iSCSI* e virtualização – sendo as duas, artificios para a utilização de *RAC 10g* em ambientes onde não esteja disponibilizado recursos de *SAN (Storage Area Network)*, requisito necessário para a implementação do produto.

Atualmente a principal riqueza de uma organização são os seus dados, e para isto, há a necessidade de que estes dados estejam armazenados adequadamente, em segurança e que haja principalmente a confiabilidade dos mesmos.

Além do aspecto da segurança dos dados, outro fator que preocupa as organizações atualmente é o fato de que estas informações estejam sempre em alta disponibilidade, ou seja, reduzindo-se ao máximo o risco destas informações ficarem por algum motivo indisponíveis aos seus usuários ou clientes, bem como uma maior escalabilidade dos servidores, uma redução do *TCO (Custo Total de Propriedade)* e conseqüentemente um melhor gerenciamento da estrutura existente.

Para garantir a disponibilidade dos dados e a segurança das operações, os *SGBD* (sistemas gerenciadores de banco de dados) atuais contam com diversos mecanismos de contingenciamento e possibilitam a integração de hardware extra, tais como áreas de armazenamento de dados do tipo *storage* (Área centralizado de armazenamento em ambiente de rede) e a criação de *grid* (processamento computacional através de vários servidores interconectados) de servidores.

O *SGBD* analisado nesta pesquisa é o *Oracle Database 10g*, o qual fornece a possibilidade de operação em *grid* através da *feature RAC*.

Como será apresentado adiante nesta pesquisa, a solução *RAC* da *Oracle* permite que mais de um servidor seja colocado em operação de forma a realizar o balanceamento de carga de processamento, aumentando com isto o desempenho e o contingenciamento em caso de falhas criando com isto um *grid* de servidores. Para isto, é necessário que os servidores acessem os dados a partir de um único ponto de armazenamento, papel este que na maioria das vezes é desempenhado por um *storage*.

O ponto único de armazenamento é justificado pelo fato de que em caso de falhas em um dos servidores, seja possível que outro servidor assuma as tarefas do servidor falho. Para que isto ocorra, os dados devem estar disponíveis ao novo servidor, normalmente adotando-se um dispositivo de *storage* para seu armazenamento.

1.1 Delimitação do Tema

O referido projeto aborda a otimização dos recursos na área de TI, especialmente referentes à distribuição de processamento e da infra-estrutura existente nas empresas e que venham a utilizar a tecnologia de *Grid Computing* através da ferramenta *Oracle RAC 10g*, com a possibilidade de implementação desta tecnologia em sistemas virtualizados ou em um ambiente que possua a tecnologia *iSCSI*. O projeto busca detalhar o que sejam estas tecnologias, bem como identificar possíveis benefícios ou problemas devido ao *Oracle RAC 10g* estar hospedado em servidores virtuais ou em ambientes de rede que possuam a tecnologia *iSCSI*. A tecnologia de *Grid Computing* desenvolvida pela Oracle visa oferecer alternativas para a racionalização de recursos tecnológicos às empresas e profissionais de TI que se deparam, por exemplo, como gargalos de processamento e as altas demandas de requisições sobre os servidores, ocasionando oneração do tempo de trabalho dos funcionários, perda de produtividade, competitividade e conseqüentemente de parte do lucro, além de propiciar alta disponibilidade e escalabilidade ao ambiente. Um requisito básico para a implementação de um sistema *RAC* é a existência de *storage*, ou seja, um ponto único de armazenamento em rede, também conhecida pela sigla SAN. Este requisito, que no projeto não será contemplado, vai obrigar a solução a ser implementada sobre as duas tecnologias abordadas neste projeto, a fim de possibilitar a instalação e configuração da ferramenta Oracle *RAC 10g* aos ambientes de testes e desenvolvimento de aplicações, sem termos os altos custos de aquisição de *storages* para tal fim. A partir deste cenário apresentado, será

desenvolvido o projeto para avaliação de desempenho de bases de dados *Oracle 10g*, através da solução de *Grid Computing* em sistemas virtualizados, bem como em sistemas que se utilizem da tecnologia *iSCSI* em sua estrutura de rede.

2 SGDB ORACLE DATABASE 10G

Neste capítulo, será abordado o tema do *Oracle Database 10g* propriamente dito, dando uma noção superficial de como é composto este banco de dados, bem como uma noção da inserção da *Oracle Corporation* no mercado mundial de *softwares* de banco de dados.

2.1 SGDB Oracle Database 10g

A *Oracle Corporation* é atualmente a líder mundial no segmento de softwares DBMS (*Sistemas de Administração de Banco de Dados*).

Segundo pesquisas divulgadas em ORACLE (2008), “de acordo com a empresa de pesquisa de mercado IDC, a Oracle é a líder no setor mundial de software de sistemas de gerenciamento de bancos de dados”. Segundo a pesquisa realizada pela IDC, na qual afirma que o percentual de 23,2% da receita mundial de software de DBMS embutido por fornecedor supera largamente o segundo concorrente mais próximo, com 14,1%. Segundo a IDC, a Oracle alcançou um índice de crescimento anual de 23,3%, quase o dobro do índice do setor como um todo, que é de 11,7%”.

Conforme afirma FARIA (2005), “A *Oracle* é atualmente o único banco de dados do mercado que possui portabilidade para praticamente todos os sistemas operacionais disponíveis no mercado. Desde dispositivos móveis com Windows CE até mainframes IBM com o OS/390, a portabilidade das informações armazenada em uma base Oracle e as características tecnológicas do produto são mantida independentemente do sistema operacional utilizado. Esta portabilidade conseqüentemente preserva os investimentos feitos pelo setor de TI, não ocorrendo desperdícios”.

A funcionalidade e o desempenho da base de dados e das aplicações são componentes importantes de sua disponibilidade. Se por exemplo, a base de dados estiver disponível, mas os usuários não estiverem conseguindo o nível do desempenho que necessitam, então os objetivos do nível de serviço e de disponibilidade não estarão sendo atingidos. A gerência e disponibilidade do banco de dados são impactadas diretamente pela forma como os recursos são distribuídos para os diversos aplicativos que acessam a respectiva base de dados.

O SGDB *Oracle Database 10g*, é um sistema no qual está composto por estruturas lógicas e físicas, sendo uma coleção de dados armazenados em um ou mais arquivos de dados, também conhecidos como *data files*. Segundo ALAPATI (2005), “a estrutura física de uma base de dados Oracle compreendem os *data files* e arquivos relacionais”. Já em sua estrutura lógica, ALAPATI (2005) enfatiza que fazem parte desta estrutura os processos *background* do banco de dados juntamente com as estruturas de memória, constituindo assim, uma instância *Oracle*, sendo responsáveis por todo o desempenho do trabalho realizado no banco de dados.

Conforme LONEY (2004), uma instância de dados *Oracle* compreende uma área de memória chamada de *SGA – System Global Area* - e os processos de *background* da própria base de dados que interagem entre a *SGA* e os *data files* presentes no disco rígido da máquina. BURLESON (2008) define *SGA* como sendo áreas de um grupo de memória compartilhada que são dedicados a uma “instância” *Oracle*, sendo uma instância definida como os programas do banco de dados e a memória *RAM* disponível na máquina.

Conforme BEST (2005), as estruturas básicas de memória juntamente com uma instância *Oracle* incluem:

- *System Global Area (SGA)*: estrutura de memória compartilhada por todo servidor e processos *background*. A *SGA* é a área de memória que contém os dados e as informações de controle da instância *Oracle*. A *SGA* está composta pelas seguintes estruturas de dados:
 - *Database buffer cache*: Cache de bloco de dados extraídos do banco de dados;
 - *Redo log buffer*: Cache das informações de *redo* (usadas para um recovery da instância) até que possa ser escrita nos arquivos físicos de redo log armazenados em disco;
 - *Shared pool*: Cache de várias construções que podem ser compartilhadas entre os usuários;
 - *Large pool*: é uma área opcional que proporciona grandes alocações de memória para certos processos, tais como operações de backup e recovery e processos de E/S do servidor.
 - *Java pool*: é usado para todas as sessões específicas com códigos java e dados que contenham máquinas virtuais java.

- *Streams pool*: é usado pelo *Oracle Streams*.
- *Program Global Area (PGA)*: é a estrutura de memória que contém dados e informações para cada processo no servidor. Um processo *Oracle* no servidor é criado a medida que um cliente fizer uma requisição ao servidor. Cada processo possui sua própria *PGA* que é criada quando o servidor inicia o processo.

BURLESON (2008) também nos esclarece que todos os processos *Oracle* utilizam a *SGA* para retenção de informações. A *SGA* é usada para armazenar dados recebidos e de informações internas de controle que sejam necessárias à base de dados. Controla-se a quantidade de memória a ser alocada para a *SGA* através da definição de alguns dos parâmetros de inicialização da base de dados.

ALAPATI (2005) ressalta que é fundamental identificarmos a diferença entre uma instância *Oracle* e uma base de dados *Oracle*, visto que muitas pessoas comumente utilizam como se fossem a mesma coisa, mas na verdade estes termos se referem a coisas totalmente diferentes.

Conforme salienta ALAPATI (2005), “uma base de dados *Oracle* é composta de arquivos, *data files* e o sistema de arquivos *Oracle*. Esses arquivos, de forma isolada são inúteis, a menos que você pode interagir com eles de certa forma, o que requer a ajuda de um sistema operacional, que fornecerá a capacidade de transformação e de recursos, tais como a memória, para permitir que se possam manipular os dados sobre as unidades de disco disponíveis no servidor. Quando se combina o conjunto de processos específicos criados pela *Oracle* sobre o servidor com a memória que lhe é atribuída pelo sistema operacional, a partir deste momento se obterá uma instância *Oracle*”. Salienta que “é comum ouvir pessoas dizerem erroneamente que o banco *Oracle* está iniciado, embora o que realmente queira se dizer é que a instância *Oracle* está iniciada, visto que o banco de dados em sua estrutura física está composto por um conjunto de arquivos e que não será possível utilizá-lo sem que a instância esteja iniciada e ativa”, complementa.

A *Oracle* usa em sua estrutura, uma sofisticada visão lógica da estrutura interna da base de dados, o que auxilia no correto armazenamento e gerenciamento dos dados nos *data files* físicos armazenados no disco rígido. Podemos separar uma base de dados em duas estruturas: física e lógica.

Segundo ALAPATI (2005), a composição lógica de uma base de dados *Oracle* está composta por:

- *Data Blocks*: É o fundamento da hierarquia de armazenamento da base de dados, sendo a base de todo o armazenamento de dados em uma base *Oracle*. Um *Data Block* consiste de um número de *bytes* alocados no disco rígido do servidor.
- *Extents*: Compreende dois ou mais *data blocks* contíguos, sendo esta a unidade de alocação de espaço.
- *Segments*: Compreende um conjunto de *extents* que será atribuída a uma estrutura lógica como por exemplo uma tabela ou índice.
- *Tablespaces*: Uma *tablespace* é um conjunto de um ou mais *data files*, e normalmente constituídos por segmentos afins. Os *data files* contém os dados de todas as estruturas lógicas que fazem parte de uma *tablespace*, como tabelas e índices.

Já conforme menciona BEST (2005), as estruturas físicas de uma base de dados *Oracle* estão assim compostas:

- *Control files*: contém dados sobre o próprio banco de dados. Estes arquivos são críticos para a base de dados, pois sem eles não pode-se abrir os *data files* para acessar os dados dentro da base de dados.
- *Data files*: Contém o usuário ou os dados da aplicação da base de dados.
- *Online redo log files*: permite a recuperação do banco de dados em caso de falhas. Se o banco de dados falhou por algum motivo e não perdeu-se nenhum *data file*, então a instância pode recuperar o banco de dados com as informações contidas nestes arquivos.

Ao utilizar uma base de dados *Oracle*, basicamente é possível criar-se em sua estrutura tabelas (*tables*) para o armazenamento dos dados e índices (*indexes*), conseguindo agilidade no tempo de resposta às consultas realizadas. Pode-se também criar sinônimos (*synonyms*) para nomes de objetos, bem como criar visões (*views*) de objetos locais ou mesmo de objetos localizados em diferentes instâncias de bases de dados através dos chamados *database link* (*dblinks*).

Conforme salienta a CRIARWEB (2008), um *dblink* é um objeto criado em um esquema de um banco de dados que possibilita o acesso a objetos de outro banco de dados, seja ele *Oracle* ou não. Esse tipo de recurso do banco de dados é conhecido como sistema de

banco de dados distribuídos, podendo ser homogêneos – quando acessam outros bancos de dados *Oracle* - e heterogêneos – quando acessam outros tipos de bancos de dados.

A estrutura básica de uma base de dados são as denominadas tabelas (*tables*). Conforme ALAPATI (2005), a base de dados *Oracle 10g* suporta vários tipos de tabelas, incluindo algumas como:

- Tabelas relacionais (*Relational tables*): São tabelas que armazenam linhas inseridas e manipuladas por aplicações. Estas tabelas são criadas através do comando *create table*.
- Tabelas organizadas por índices (*Index-organized tables*): Podem ser criadas tabelas para armazenar dados juntamente com uma estrutura de índices, agilizando o tempo de resposta de consulta aos dados da tabela.
- Tabelas externas (*External tables*): Pode-se ter acesso a diversos arquivos externos como se estes fossem tabelas dentro da base de dados. Possui algumas limitações como não poder atualizar ou excluir registros, bem como não poder criar índices para as consultas.
- Tabelas particionadas (*Partitioned tables*): Uma tabela pode ser particionada em múltiplas partições. Este recurso contribui para um melhor desempenho nas consultas efetuadas pelos usuários do banco, pois pode-se particionar uma tabela por intervalos de valores, uma lista de valores ou a combinação destas opções.
- Visões materializadas (*Materialized views*): As visões materializadas são uma réplica dos dados retornados por alguma consulta em bases de dados diferentes. Conhecida também como *snapshot*.
- Tabelas temporárias (*Temporary tables*): As tabelas temporárias podem ser criadas para que vários usuários a utilizem, porém, cada usuário verá somente os seus registros na tabela.
- Tabelas clusterizadas (*Clustered tables*): Se houverem duas tabelas que sejam consultadas em conjunto, há a possibilidade de se armazenar fisicamente em conjunto através de uma estrutura chamada de *cluster*.

Segundo ALAPATI (2005), as tabelas podem conter um único atributo ou um conjunto de atributos que podem funcionar como uma "chave". As chaves (*keys*) possuem muitas funções importantes, possuindo estas duas classificações: Chaves primárias (*Primary key*) e

chaves estrangeiras (*foreign key*). As chaves primárias atribuem um valor único para linhas da tabela, identificando o registro dentro da tabela, fornecendo com isto a forma de criação de relacionamento com outras tabelas através das chaves estrangeiras de outras tabelas. Elas são comumente usadas para unir ou combinar dados de duas ou mais tabelas. As chaves também são críticas para a criação de índices, facilitando a rápida recuperação de dados de grandes tabelas.

Para suporte das consultas as tabelas, pode-se usar as chamadas visões (*views*) nas quais limitam o número de linhas retornadas ou alteram as colunas mostradas, retornando somente os dados que se deseja em uma consulta.

Estas *views* podem ser somente leitura (*read-only*) ou atualizáveis e podem referenciar tanto tabelas locais como tabelas remotas.

Para objetos que requeiram o armazenamento físico em disco, a *Oracle* disponibilizou espaços dentro das chamadas *tablespaces*. Conforme esclarece BEST (2005), “uma base de dados está dividida em uma ou mais unidades lógicas de armazenamento chamadas *tablespaces*”.

Segundo LONEY (2004), as *tablespaces* consistem em um ou mais arquivos de dados, porém, um arquivo de dados pode pertencer somente a uma *tablespace*. Para o armazenamento de objetos lógicos, foram disponibilizadas na versão 10g duas *tablespaces* – *SYSTEM* e *SYSAUX* – responsáveis pelo dicionário de dados da base.

Somente a partir da versão *Oracle Database 10g*, há a opção de se criar um tipo especial de *tablespace* denominada “*bigfile tablespace*”, cujo tamanho pode ser superior a muitos *terabytes*.

Conforme LONEY (2004), as *tablespaces* podem ser administradas tanto por um dicionário de dados ou geridas localmente. Em uma *tablespace* gerida por um dicionário de dados, a gestão do espaço é gravada no dicionário de dados do banco de dados. Em uma *tablespace* administrada localmente - padrão no *Oracle Database 10g* - a *Oracle* mantém um bitmap em cada *datafile* da *tablespace*, o qual disponibiliza espaço disponível da mesma. Apenas as quotas de espaço são geridas pelo dicionário de dados, reduzindo drasticamente a contenção das tabelas do dicionário de dados.

2.2 Tipos de bases Oracle Database 10g

As aplicações necessitam ser desenvolvidas, testadas e só após estes passos ser colocadas em produção. Uma empresa possui normalmente três tipos de bases de dados em sua estrutura: bases de desenvolvimento, bases testes e bases de produção.

Normalmente estas bases estão alojadas em ambientes separados, embora em empresas de pequeno a médio porte sejam utilizadas estas versões de teste e desenvolvimento dentro de uma única base de dados.

Segundo ALAPATI (2005), “as bases de desenvolvimento (*Development databases*) são geralmente utilizadas pela equipe de desenvolvimento da organização, onde estas pessoas possuem o pleno acesso para acessarem e modificarem dados e objetos no banco de dados.

Já as bases de testes (*Test databases*), são utilizadas de forma a simular o ambiente de produção da organização, sendo utilizadas também para a funcionalidade dos códigos após sair do ambiente de desenvolvimento.

Somente após a seqüência destas etapas, entra em cena as chamadas bases de produção (*Production databases*). “Um novo código desenvolvido somente poderá ser implementado em um ambiente de produção se já houver sido testado com sucesso em bases de teste” complementa ALAPATI (2005).

Porém, cabe ressaltar que quando uma nova aplicação é desenvolvida e testada, sendo após colocada em produção, o ciclo de desenvolvimento e de produção não termina. Os aplicativos estão constantemente a serem alterados nas organizações em virtude de dois motivos principais:

- Correção de erros anteriores;
- Melhor funcionalidade do aplicativo.

Conforme alerta ALAPATI (2005), além dos motivos citados acima, há o papel dos usuários dos sistemas que constantemente requisitam modificações nos aplicativos com o intuito justamente de melhorar a funcionalidade dos mesmos.

2.3 Desempenho e tolerância a falhas em bases *Oracle Database 10g*

O *SGDB Oracle 10g* apresenta recursos que melhoram significativamente o desempenho de uma base de dados, além de fornecer uma tolerância a falhas através do recurso de clusterização, feita pela *feature RAC - Real Application Cluster* - presente na versão 10g do *SGDB*.

Dentre as melhorias de desempenho possíveis, podemos citar algumas como índices (*index*), particionamento de tabelas (*Partitioned tables*) cujo tamanho seja elevado e também o uso de estatísticas nas mesmas.

Os índices (*indexes*), segundo ALAPATI (2005), proporcionam um acesso rápido às linhas das tabelas, ordenando os valores por suas colunas específicas, usando estes valores ordenados para procurar facilmente a linha da tabela associada.

Índices permitem encontrar uma linha com um certo valor coluna sem ter de olhar para mais do que uma pequena fração do total de linhas de uma tabela. Assim, a utilização adequada dos índices reduz o I/O de disco rígido para níveis mínimos. Os índices são estruturas de dados meramente facultativas, estando totalmente mantidos pelo *SGDB*.

Conforme LONEY (2004), para agilizar a consulta às tabelas da base de dados, a Oracle disponibiliza suporte a vários tipos de índices (*indexes*), como por exemplo:

- *B*- tree indexes*: é o tipo de índice padrão disponível no Oracle, sendo muito útil para a seleção linhas que satisfazem os critérios de equivalência ou de uma de uma série critérios.
- *Bitmap indexes*: para colunas que possuem poucos valores únicos, um índice bitmap pode ser capaz de melhorar a performance consulta. Porém, só deve ser utilizado quando os dados são carregados em lote (como em muitos *Data Warehouse*).
- *Reverse key indexes*: se houverem conexões de I/O (entrada e saída) durante as inserções seqüenciais dos valores, a base de dados Oracle pode dinamicamente inverter os valores indexados antes de armazená-los.

- *Partitioned indexes*: você pode particionar índices para dar suporte às tabelas particionadas ou para simplificar a manutenção de índices. O particionamento de índices pode ser local ou ser aplicáveis globalmente a todas as linhas na tabela.
- *Text indexes*: você pode indexar valores de texto para apoiar o aumento da capacidade de busca, tais como a expansão da palavra derivada ou a procura por frases. Os índices de texto são o conjunto de tabelas e índices mantidos pela Oracle para apoiar as complexas exigências nas buscas de texto.

Dentre as melhorias possíveis citadas, encontramos o recurso de particionamento de tabelas (*partitioned tables*). ALAPATI (2005) informa que em certos casos, “bases de dados Oracle podem ter um tamanho muito grande, e não é raro encontrar casos em que tabelas de dados cujos tamanhos chegam a vários *gigabytes*. O particionamento é uma forma de dividir logicamente uma grande tabela de dados em pequenos pedaços para facilitar consultas e conseqüentemente sua administração”.

Embora tabelas particionadas geralmente busquem melhorar o desempenho em tabelas muito grandes, elas não são a solução para os problemas de codificação ou outros problemas de design das aplicações existentes. Porém, o particionamento se tornará moroso em relação ao trabalho adicional que será despedido para a manutenção das partições e seus índices.

Conforme salienta ALAPATI (2005), o particionamento de tabelas também é uma forma eficaz de excluir ou arquivar os dados mais antigos que não são necessárias naquele momento. Exemplo disto são os *Data Warehouses* que guardam os dados que forem mais antigos que uma certa data, arquivando-os em tabelas particionadas facilmente.

Entende-se por estatísticas (*statistics*), o processo de estimativas para melhorar os acessos às tabelas da base de dados. Conforme ALAPATI (2005), a partir da versão *Oracle Database 10g*, a Oracle passou a permitir a coleta automática de estatísticas, todavia também a execução de estatísticas poderá ser realizada manualmente.

Como ressalta ALAPATI (2005), devem-se coletar manualmente estatísticas sob as seguintes condições:

- Quando utiliza-se de tabelas externas;
- Quando necessite coletar estatísticas do sistema;
- Para coletar estatísticas sobre objetos fixos, ex. tabelas;

- Imediatamente após executar algum processo que faça cargas em tabelas, desatualizando os dados estatísticos das mesmas.

Um comparativo quanto à utilização de estatísticas em tabelas com tabelas que não as possuam, concluem que tabelas com estatísticas são entre 30 por cento e 50 por cento mais rápidas, por vezes, em termos de desempenho, salienta ALAPATI (2005).

Um ponto também muito relevante hoje em dia nas organizações, diz a respeito à tolerância a falhas a que seus dados estão propensos. Abaixo, a figura 1 demonstra por setores da indústria americana, o impacto financeiro por hora não trabalhada devido à indisponibilidade de sistemas informatizados.

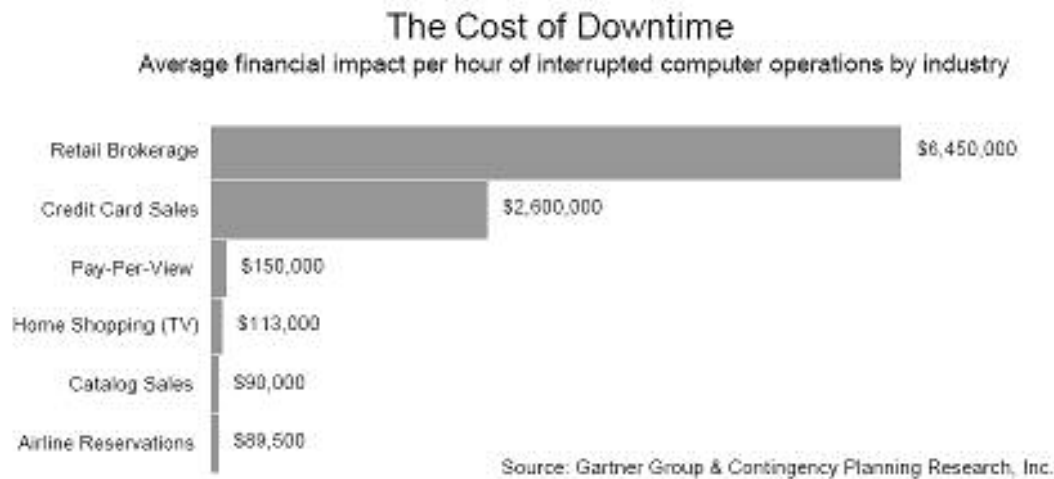


Figura 1. Custos de *downtime*. THOME(1999)

Conforme salienta THOME (1999), ao calcular os custos de inatividade (*downtime*) para o negócio, não pode-se esquecer de incluir tanto os custos diretos quanto os indiretos. Os custos diretos são o mais fácil de calcular. Estes incluem as receitas perdidas que estejam diretamente relacionados com a inatividade dos sistemas. No entanto, também pode-se incorrer substancialmente nos custos indiretos em decorrência desta inatividade. Estas são situações como, por exemplo, a perda do bom relacionamento dos clientes em relação a empresa, ou a má impressão gerada pela falta da cobertura dos sistemas. A organização com isto, poderá perder não somente os negócios atuais, mas também negócios futuros.

Proporcionar um elevado nível de disponibilidade banco de dados não é tão simples, nem barato. No entanto, há muitos casos em que houve sucesso nas soluções que atendessem os exigentes requisitos da alta disponibilidade. O desafio é entender o que pode estar errado,

para analisar os custos de *downtime* - tempo relativo à inatividade de sistemas informatizados - esperados que estejam associados a esse problema, e comparar estes custos contra os custos das várias soluções que minimizam estes problemas associados ao *downtime*, completa THOME(1999).

Dentre as causas não planejadas de um *downtime*, estão inclusas falhas de hardware, falhas de softwares, erros humanos e desastres em geral. Segundo THOME (1999), um dos verdadeiros desafios na concepção de uma solução altamente disponível está em examinar e abordar todas as possíveis causas de uma possível inatividade, sendo importante considerar todas as possíveis causas ou imprevistos do *downtime*. Na figura 2, apresenta-se um diagrama das falhas inesperadas, classificando-as em falhas de software, falhas de hardware, erros humanos, e desastres. Em cada classificação é apresentada uma lista de possíveis causas de falhas relacionadas a essa categoria.

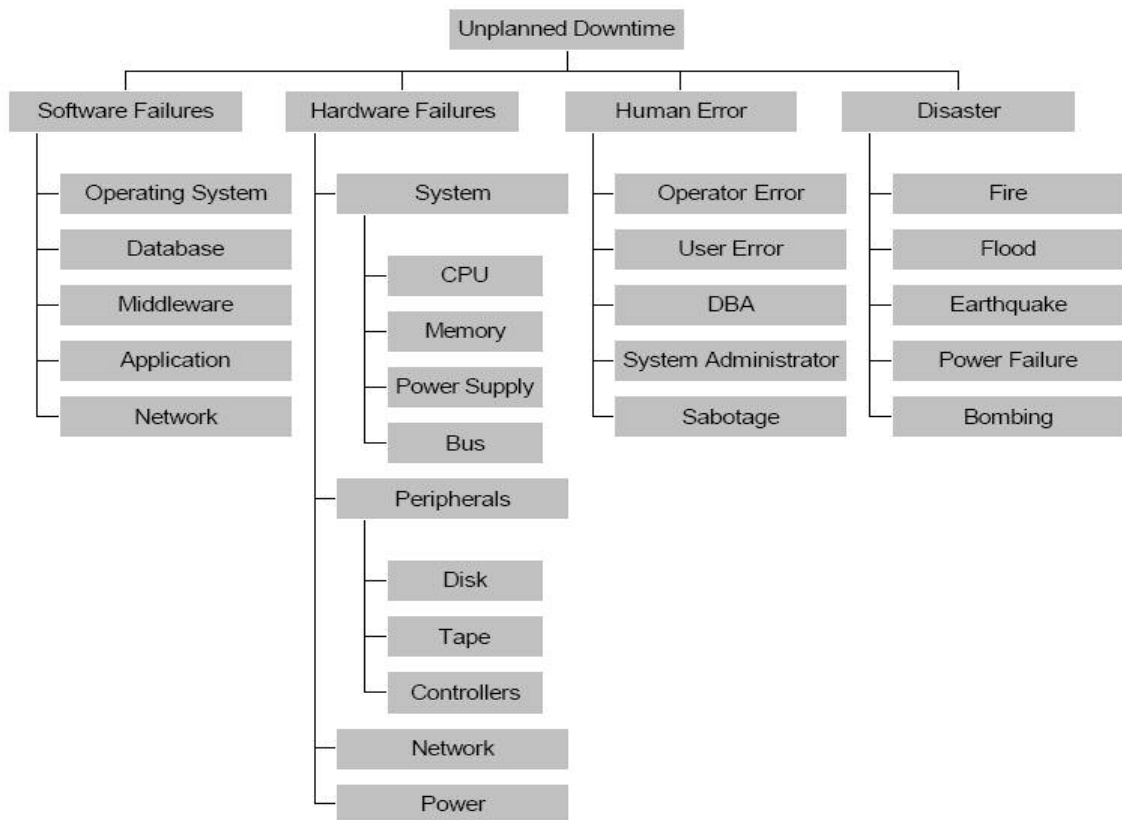


Figura 2. Possíveis causas não planejadas de *downtime*. THOME(1999)

Como verifica-se na figura acima, a categoria das possíveis falhas em softwares englobam falhas do sistema operacional, banco de dados, *middleware*, aplicações, e as

possíveis falhas na rede de computadores. A falha de qualquer um destes componentes pode resultar em uma parada do sistema.

Já na categoria das possíveis falhas de *hardware* estão inclusas as falhas de sistema, periféricos, rede e faltas de energia elétrica. Os erros humanos - uma das principais causas de falhas - incluem os erros relativos aos operadores, erros de usuários do sistema, erros dos administradores de bancos de dados, administradores de sistema ou até mesmo sabotagem. A última categoria de *downtimes* não planejados são as catástrofes ou desastres naturais. Embora sejam raras, estas causas de inatividade poderão ter extremo impacto sobre as empresas, devido ao seu efeito prolongado sobre as operações da organização. Dentre as possíveis causas dos desastres estão listados os incêndios, inundações, terremotos, raios e bombas. Uma solução de alta disponibilidade bem concebida terá que levar em conta todos estes fatores na prevenção de inatividades não planejadas.

2.4 Bases Oracle Database 10g *Stand-alone*

Entende-se por bases *stand-alone*, as bases que funcionam como se fossem um espelho da base Oracle principal atualizadas através dos *archive log* gerados pela base principal. Os *archive logs* são o conjunto de comandos executados pela base principal e armazenados em um único arquivo. Segundo ASHDOWN (1999), “uma base de dados *stand-alone* é um banco de dados criado a partir de uma réplica do banco de dados primário através de um backup conseguido através dos *archive logs*”. A base *stand-alone* (ou também chamada de base de espera) ao aplicar os *archive logs* gerados a partir da base principal, se mantém sincronizada ao refazer todos os comandos gravados nos *archive logs* recebidos.

A figura 3 mostra a estrutura de funcionamento e sincronização entre uma base primária e uma base *stand-alone*. A partir do diagrama abaixo, nota-se que a base *stand-alone* é uma base de dados normalmente em estado de recuperação (*recovery*), aplicando os *archive logs* gerados pela base de dados principal. Quando a base *stand-alone* está ativada, ele passa a ser a base de dados de produção e já não sendo mais compatível com a base de dados principal.

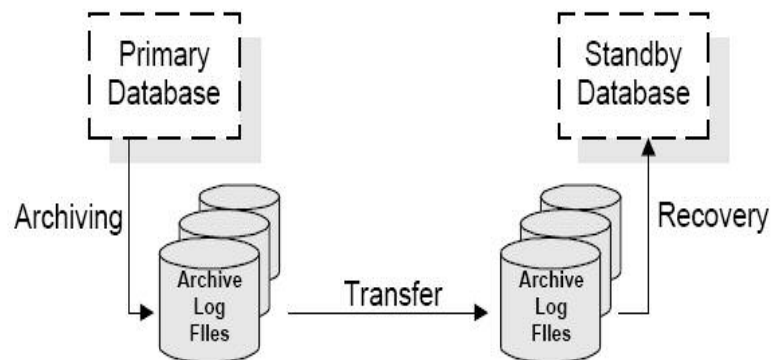


Figura 3. Estrutura do funcionamento entre base primária e base *stand-alone*. TO (1999)

Conforme destaca TO (1999), “A base principal é uma base de dados que contém dados de produção. Um banco de dados de espera – *stand-alone* - é uma cópia do banco de dados primário alavancada essencialmente para recuperação de desastres”.

Segundo salienta ASHDOWN (1999), uma base de dados *stand-alone* se propõe a cumprir os seguintes objetivos:

- Proteção contra desastres;
- Proteção contra corrupção de dados;
- Relatórios complementares.

Com este cenário, se por ventura a base principal sofrer algum tipo de avaria ou seus dados forem corrompidos, pode-se recuperar a base de dados *stand-alone* para que esta assuma o papel da base principal.

2.5 Bases Oracle RAC 10g

A computação em *grid* desenvolvida pela *Oracle* criou uma infra-estrutura de software que faz o balanceamento de todos os tipos de requisições em todos os servidores do cluster e permite que todos estes servidores seja administrados por um sistema completo.

Segundo BEST (2005), a tecnologia de computação em *grid* desenvolvida pela *Oracle* inclui:

- *Automatic Storage Management (ASM)*: cria e mantém uma rede de armazenamento, fornecendo um alto nível de administração com baixos custos. Quando discos são adicionados ou retirados, o *ASM* redistribui automaticamente os dados entre os discos remanecentes.
- *Real Application Clusters (RAC)*: executa todas as requisições em um *cluster* de servidores.
- *Oracle Streams*: proporciona um enquadramento unificado para o compartilhamento de informações, combinando filas de mensagem, replicação de dados, notificação de eventos, cargas de *Data Warehouses* em uma única tecnologia.
- *Enterprise Manager Grid Control*: gerencia as operações do *grid*, incluindo a administração de toda a relação de softwares, provisionamento usuários, clonagem de bases de dados e gerenciamento de *patches*.

Conforme FARIA (2005), “a idéia central do *Grid Computing* é oferecer a computação como um serviço público. Não deve-se ter preocupação com o local onde os dados residam ou qual computador processe a solicitação. O usuário deve ser capaz de solicitar – e receber – informações ou recursos de computação no volume e frequência que desejar. Pode-se fazer uma analogia com o modo como funcionam os serviços públicos elétricos: você não sabe onde está o gerador, nem como é a rede elétrica. O que se deseja é eletricidade. E a consegue. O objetivo é tornar a computação um serviço público e onipresente”.

Grid computing é uma arquitetura relativamente nova que remete ao problema das ilhas de computação. FARIA (2005) define *grid computing* como o “uso coordenado de muitos usuários pequenos que agem como um grande computador para rodar aplicações de uma organização”.

A infra-estrutura construída pela TI ao longo dos anos criou as chamadas ilhas de computação. Cada aplicação da empresa tem seus próprios servidores e *storages* dedicados. Como as demandas da aplicação mudam durante o tempo, um servidor pode estar com utilização máxima enquanto um outro servidor pode estar sendo sub-utilizado. Para combater estes desperdícios, os clientes compram capacidade extra de servidores e de armazenamento para atender a demanda de pico.

Conforme FARIA (2005), “o *hardware* é dimensionado para a carga de pico de uma determinada aplicação, isso geralmente significa que em algumas aplicações a utilização do *hardware* pode ser mínima enquanto em outras a utilização do *hardware* é maximizada ao extremo”.

De fato, pesquisas apontam que a média da indústria para a utilização de servidores é de aproximadamente 30%. Isso significa muita capacidade computacional desperdiçada. E não apenas o *hardware*. “*Software* e mão-de-obra custam mais, uma vez que eles devem ser dimensionados pelo pico. *Hardware* (21%), o *software* (24%) e o trabalho do administrador de sistemas ou DBA (40%) consomem 85% do orçamento típico do TI”, ressalta FARIA (2005).

Com este cenário, não há a possibilidade da escalabilidade nesta arquitetura. Quando as empresas alcançam o limite da capacidade de seus servidores, necessitam comprar um novo servidor para rodar sua aplicação, com isto, acabam arcando com gastos consideráveis.

“Também do ponto de vista da disponibilidade, aplicações executadas em um servidor simples significam um ponto único de falha, a menos que as companhias tenham um servidor de *backup* disponível. Assim, o *uptime* - tempo de sincronização necessário após um servidor ser reiniciado - ou disponibilidade não é ideal nestes ambientes distribuídos”, complementa FARIA (2005).

Com todos estes aspectos mencionados, a segurança também será afetada, visto que cada sistema deve manter sua própria infra-estrutura de segurança. As ilhas individuais resultam inevitavelmente em várias senhas, geralmente com políticas inconsistentes de segurança e em níveis variados de conhecimento para a gerência da segurança dos sistemas.

A computação em *grid* oferece muitas vantagens, entre elas destacamos:

- Quando a infra-estrutura existente tem ilhas de servidores, um *grid* corporativo agrupa os recursos juntos e compartilha aqueles recursos entre as aplicações;
- A infra-estrutura existente tem altos custos de *hardware* e *software* devido à complexidade e ineficiência. Um *grid* pode ser construído usando-se componentes de baixo custo, tais como servidores blade Intel, sistema operacional Linux e armazenamento modular. A maior razão pela qual o custo do software diminui é que a utilização do hardware (por processador) aumenta.

- A infra-estrutura existente pode ser difícil de reconfigurar conforme as cargas mudam. Com um *grid*, os recursos podem ser alocados quando são necessários por causa de mudanças de carga (balanceamento de cargas automático).
- A infra-estrutura existente é difícil de gerenciar porque cada sistema é único. Com um *grid* corporativo, todos os sistemas são padronizados e gerenciados como um só recurso;
- Qualidade de serviços de missão crítica nos padrões industriais, utilizando servidores de baixo custo;
- Ser tolerante à falhas;
- Capacidade sob demanda;

A tecnologia de *Grid Computing* visa oferecer alternativas para a racionalização de recursos tecnológicos às empresas e profissionais de TI que se deparam, por exemplo, como gargalos de processamento e as altas demandas de requisições sobre os servidores, ocasionando oneração do tempo de trabalho dos funcionários, perda de produtividade, competitividade e conseqüentemente de parte do lucro.

A tabela abaixo, evidencia algumas situações problemáticas enfrentadas atualmente pelas áreas de TI, bem como as soluções oferecidas pela solução *Grid* para resolvermos tais situações.

Tabela 1. Problemas TI vs Soluções *grid*

Problemas de TI	Soluções com tecnologia <i>Grid Computing</i>
Servidores isolados	Servidores compartilhados
Altos custos com hardware e software	Componentes de baixo custo
Configurado pelo pico	Capacidade conforme demanda
Ponto único de falha	Tolerância a falhas
Difícil mudança	Flexibilidade
Difícil gerenciamento	Gerenciamento unificado

Assim sendo, a solução *Oracle RAC 10g* se apresenta como a um método eficaz para evitar que a falha de um nó do cluster ocasione a indisponibilidade da aplicação, tornando as aplicações mais escaláveis e altamente disponíveis.

Conforme destaca FARIA (2005), um dos principais benefícios desta tecnologia é a possibilidade de utilizarem-se servidores padrão, de baixo custo como (Intel/Linux, Blades, etc.), garantindo altos níveis de serviços e ao mesmo tempo a redução dos custos de infraestrutura.

No *Oracle 10g*, o banco de dados pode começar imediatamente a executar o balanceamento da carga de trabalho em um novo nó com a nova capacidade de processamento à medida que esse nó é provisionado novamente de um banco de dados a outro, além de liberar uma máquina quando esta não for mais necessária.

Isso é o que se chama de capacidade sob demanda. Outros bancos de dados não conseguem se expandir e diminuir durante a execução e, portanto, não conseguem utilizar o hardware com a mesma eficiência.

FARIA (2005) ressalta que um ponto muito importante disponibilizado em um sistema *RAC* é a escalabilidade. Este aspecto torna o poder de processamento incremental imediatamente disponível para o trabalho, não necessitando de reparticionamentos ou redistribuições de dados.

Abaixo, a figura 4 nos dá a idéia de como novos servidores adicionados ao cluster tem acesso imediato aos dados existentes.

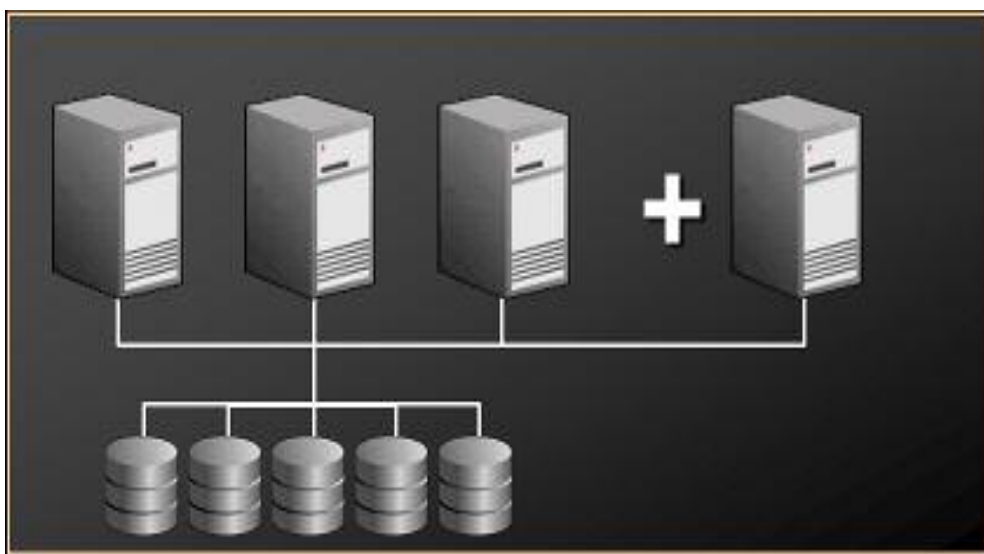


Figura 4. Estrutura de escalabilidade ilimitada em *RAC 10g*. FARIA (2005)

Geralmente as instâncias *RAC Oracle 10g* – também conhecidas por “nós” – estão localizadas em servidores separados entre si e interconectados através de uma estrutura de rede de alta velocidade para que se consiga um desempenho ideal. Esta estrutura de rede pode ser uma estrutura de *fiber channel* – fibra ótica – ou até mesmo a utilização da própria estrutura de rede existente na organização se utilizarmos a solução de *iSCSI* para a implementação do *Oracle RAC 10g*, como será visto mais adiante em uma das alternativas apresentadas na pesquisa.

Um aspecto a ser ressaltado é o fato de um *RAC* fornecer níveis elevados de disponibilidade para aplicações de missão crítica. Em um ambiente *RAC*, o *Oracle* funciona em dois ou mais sistemas através de um cluster, com acesso simultâneo a uma única base de dados compartilhada armazenada em uma estrutura de armazenamento de rede, ou seja, em uma estrutura de *storage*. No caso de uma falha de um dos sistemas, os sistemas remanescentes não afetados pela falha executam a recuperação da instância do *Oracle* que falhou. Isto fornece benefícios de disponibilidade e escalabilidade se comparados a uma solução de *failover* a frio.

As falhas de um dos sistemas afetam somente um subconjunto de seus usuários. A falha não afeta os usuários conectados a outros sistemas no cluster. Os usuários do sistema que falhou podem ser reconectados nos sistemas sobreviventes do cluster. Este *failover* é mais rápido, porque o servidor *Oracle* já está ativo nos servidores que não falharam.

Uma vez que todos os servidores estejam ativos durante a operação normal, os recursos do sistema são melhor utilizados, e as cargas pesadas podem ser sustentadas.

Além disso, esta solução também apresenta uma maior escalabilidade operacional. Se uma eventual capacidade adicional for necessária, novos servidores adicionais podem ser incorporados ao cluster.

Segundo definição de DYKE (2006), “um *grid* é constituído por vários servidores interligados que são exibidos aos usuários finais e aplicações como se fossem um único servidor”. Ressalta DYKE (2006) que “um sistema em *RAC* compreende dois ou mais nós físicos, sendo possível que um único nó dentro de um cluster *RAC* possa falhar sem comprometer a disponibilidade do cluster como um todo”, caracterizando perfeitamente as vantagens de alta disponibilidade e escalabilidade oferecidas pela tecnologia *RAC 10g*.

Assim sendo, salienta DYKE (2006), “uma base de dados *RAC* permite múltiplas instâncias de dados residentes em servidores diferentes do cluster acessarem uma base de

dados comum localizada em um *storage*”. Abaixo, a figura 5 ilustra a arquitetura de uma estrutura *Oracle RAC 10g*. Nesta figura, nota-se a criação do *grid* de servidores, os quais acessam a base de dados armazenada em um *storage* compartilhado. O *storage* é compartilhado através de um enlace denominado *Storage Network*.

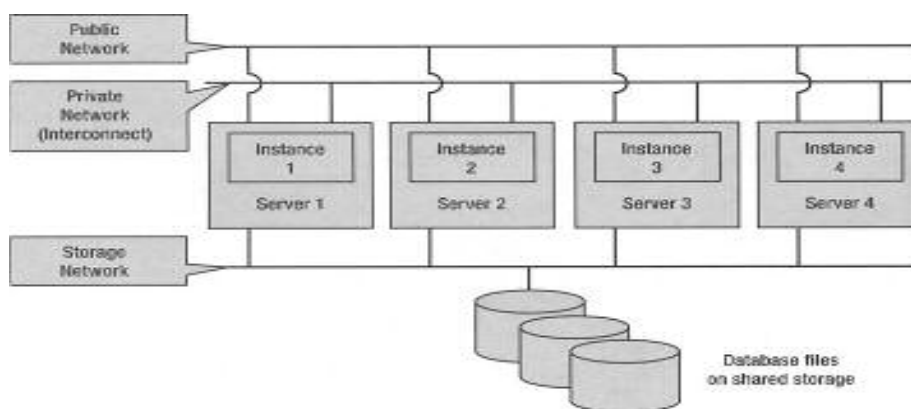


Figura 5. Estrutura de um cluster RAC com 4 nós. DYKE (2006)

Conforme ressaltado anteriormente, o projeto visa a implementação de um ambiente *Oracle RAC 10g*, buscando alternativas ao requisito de *storage* necessário originalmente para a referida implementação. Tendo em vista que o valor de mercado de um equipamento de armazenamento de dados em rede – *storage* – é elevado, sendo assim esta realidade se torna impeditiva e faz com que se torne restrita a utilização da ferramenta de clusterização em bases de dados *Oracle* em ambientes de testes ou desenvolvimento.

2.6 Licenciamento

Um fator a ser levado em consideração para a implementação de um futuro projeto que envolva a utilização de *RAC Oracle 10g* é relativo ao preço das licenças de utilização do produto. Pensando neste aspecto, foi realizada uma pesquisa na qual chegou-se a uma média de preços praticados pelo mercado, bem como as modalidades de licenciamento de um *RAC 10g*.

O produto *Oracle RAC 10g* possui basicamente três modalidades de licenciamento:

- *Standard Edition*
- *Enterprise Edition*

➤ por usuários

A versão *Standard Edition* se baseia no fato da quantidade de processadores a ser utilizada pelos nós do *RAC*. Esta possibilidade de licenciamento está limitada á utilização de até 04 processadores. Este modo de licenciamento não contempla suporte *Oracle* nativo a não ser pelo pagamento de um percentual anual que gira em torno de 20%. O custo estimado por processador gira em torno de R\$ 65.000,00. Supondo que um *RAC Oracle* possua 02 nós bi-processados, ou 4 nós mono-processados, o custo total de uma solução de base de dados clusterizada seria em torno de R\$ 260.000,00.

A versão *Enterprise Edition* também baseia-se no fato da quantidade de processadores presentes na estrutura do *RAC*, contudo parte do principio de termos mais de 4 processadores e sem limite máximo dos mesmos. Os valores variam de acordo com o modelo de processador Esta opção de licenciamento contempla suporte *Oracle* irrestrito. O custo estimado em média por processador gira em torno de R\$ 220.000,00.

Já a versão de licenciamento por usuários, não foi possível obter um levantamento em relação aos valores das licenças, devido a ser um tipo de licenciamento muito pouco utilizado, pois é extremamente raro empresas contratarem licenciamentos definindo um número exato de funcionários que irão utilizar o banco de dados, geralmente o que se pretende em uma organização quando se adota uma ferramenta como estas de clusterização de bases de dados, é que se tenha um número grande de pessoas utilizando o mesmo e não com um número limitado de acessos.

3 AMBIENTES DE ARMAZENAMENTO DE DADOS PARA BASES ORACLE RAC 10G

Para realizar o armazenamento de dados em uma estrutura *RAC 10g*, a condição necessária para a implementação desta solução passa invariavelmente pela utilização de um ponto único de armazenamento em rede, conforme dito em outras ocasiões, papel desempenhado por um *storage*.

O ponto único de armazenamento em rede é justificado pelo fato de que em caso de falhas em um dos servidores, seja possível que outro servidor assuma as tarefas do servidor

falho. Para que isto ocorra, os dados devem estar disponíveis ao novo servidor através do acesso ao *storage* disponibilizado.

Os ambientes de armazenamento disponíveis para a implementação desta pesquisa que visa à implementação da ferramenta *Oracle RAC 10g* estão dispostos da seguinte maneira:

- *Storage*;
- Ambiente virtual;
- *iSCSI*

Para esta pesquisa, serão utilizados dois meios de armazenamento alternativos ao *storage* sendo estes, alternativas para a implementação da computação em *grid* através do *RAC* conforme mencionados acima.

3.1 *Storage*

Storage é um meio físico pelo qual estão dispostos em um único ponto central, todo o armazenamento dos dados. Seu uso é justificado pelo fato de que em caso de falhas em alguns dos servidores do *RAC*, seja possível que outro servidor assuma as tarefas do servidor falho. Para que isto ocorra, os dados devem estar disponíveis ao novo servidor, normalmente adotando-se um dispositivo de *SAN - Storage Area Network* - para seu armazenamento.

Uma estrutura de rede baseada em arquitetura *SAN* tem por finalidade proporcionar uma infra-estrutura lógica e física para a transferência dos dados entre as aplicações de sistema e os dispositivos de armazenamento.

Conforme exemplifica FONSECA (2008), “a arquitetura *SAN* é uma infra-estrutura de rede dedicada ao compartilhamento de dispositivos de armazenamento para servidores de aplicação, proporcionando flexibilidade, alta disponibilidade e escalabilidade para os sistemas corporativos, bem como o armazenamento de todas as informações dos servidores de aplicação em um único ponto de armazenamento de dados”.

Abaixo na figura 6, um exemplo ilustrando a arquitetura de armazenamento *SAN*.



Figura 6. Arquitetura Redes de Armazenamento SAN.FONSECA(2008)

Porém, cabe ressaltar que o valor de mercado de uma *SAN* é um valor razoavelmente elevado, girando entre R\$ 35.000,00 até R\$ 300.000,00, tornando quase impeditivo para empresas de pequeno porte ou instituições de ensino que necessitem utilizar alguma tecnologia que necessariamente esteja atrelada à utilização de uma *SAN*.

Este seria o cenário ideal para a implementação da clusterização de bases Oracle através da *feature RAC 10g*.

3.2 Ambiente Virtual

A primeira solução proposta na pesquisa é o recurso de virtualização, baseando-se na utilização de virtualização de *hardware* através da ferramenta *VMware Server*. Através desta ferramenta serão criados dois servidores virtualizados dentro de um único servidor físico. Estes servidores virtuais possuirão sistema operacional *Oracle Enterprise Linux*. Em cada servidor será configurada uma instância *RAC* sendo possível a criação de uma área de armazenamento comum para os servidores, atuando como se fosse um emulador do *storage*, vindo a ser utilizada esta emulação na implementação da ferramenta *Oracle RAC 10g* sobre o sistema virtualizado.

Conforme SYNGRESS (2008), “o planejamento e a implementação de um servidor virtualizado é um processo complexo”, visto que depende de uma análise criteriosa do que se pretende utilizar com o novo servidor virtual, quais aplicativos estarão presentes e sua carga de utilização para que se possa dimensioná-lo dentro do aplicativo de virtualização.

Segundo SHIELDS (2008), “a virtualização pode mudar muito o modo de operar e gerenciar seu ambiente de TI se você implementar e geri-lo corretamente. Com as corretas considerações dos sistemas de inventário e de desempenho, você pode comprimir dezenas de servidores em um único *host*” conseguindo com isto, uma redução de espaço físico destas máquinas, bem como uma redução dos custos com energia e manutenções de hardware.

Abaixo, a figura 7 apresenta o diagrama da estrutura *Oracle RAC 10g* através da utilização do recurso da virtualização, com o *RAC* contendo 02 nós.

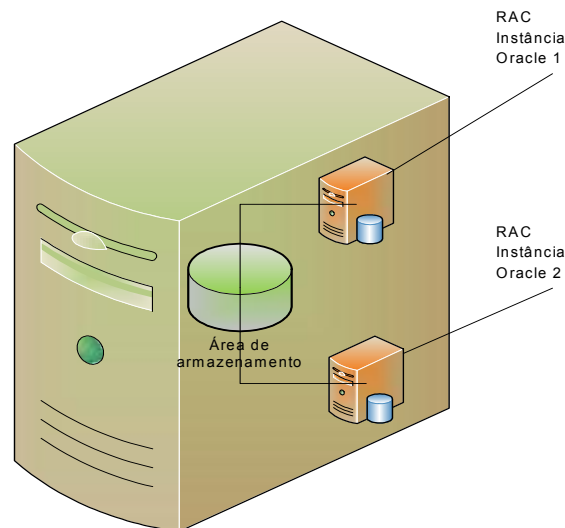


Figura 7. Diagrama da arquitetura Oracle RAC 10g através de virtualização

3.3 *iSCSI*

A segunda solução é a adoção da tecnologia *iSCSI*, sendo esta uma alternativa ao protocolo *SAN* por meio de um conjunto de comandos *SCSI* sobre o protocolo de rede *TCP/IP* em redes ethernet já disponíveis no ambiente das organizações.

Conforme descreve FONSECA (2008), “*iSCSI* é um projeto multi-plataforma que permite transferências de dados em alta performance. O *iSCSI* - Internet *SCSI* (*Small*

Computer System Interface) é baseado em *storage* para rede, facilitando o armazenamento de dados. É também um sistema *SCSI* trabalhado em cima de uma rede TCP/IP, o que facilita a transferência de dados sobre redes”.

Como destaca FONSECA (2008), o principal objetivo do protocolo *iSCSI* é prover uma camada de transporte para a camada *SCSI*, que permite a comunicação entre os dispositivos de E/S nos sistemas de armazenamento.

Já HOWTOFORGE (2008) destaca que o *iSCSI* trata-se de uma rede de armazenamento que atua sobre o protocolo *TCP / IP*, encapsulando dados *SCSI* em pacotes *TCP*.

A estrutura *iSCSI* nos permite contatar um *host* para servir um *array* de armazenamento através de uma simples conexão de rede *ethernet*, sendo esta a solução mais barata do que uma estrutura *Fibre Channel* onde os equipamentos geralmente são caros, destaca HOWTOFORGE (2008).

Os dispositivos *iSCSI* não devem ser confundidos com os dispositivos *NAS – Network Attached Storage* – como por exemplo *NFS (Network File System)*. A diferença mais importante é que volumes *NFS* podem ser acessados por várias máquinas simultaneamente, porém volumes *iSCSI* podem ser acessados por somente um *host*, conforme esclarece HOWTOFORGE (2008).

Alguns críticos alegam que a estrutura *iSCSI* tem um desempenho pior em comparação com estruturas *Fibre Channel* e que provocaria altas cargas de CPU na máquina *host*. Obviamente que o desempenho não pode ser comparado a uma estrutura *Fibre Channel*, pois os custos com equipamentos são totalmente diferenciados. Todavia, se a alternativa de *iSCSI* estiver contemplada por uma estrutura *ethernet gigabit*, o desempenho geralmente seria suficiente.

Abaixo, a figura 8 apresenta a pilha de protocolos utilizados para a transferência de comandos *SCSI* em uma rede *TCP/IP* sobre *ethernet*.

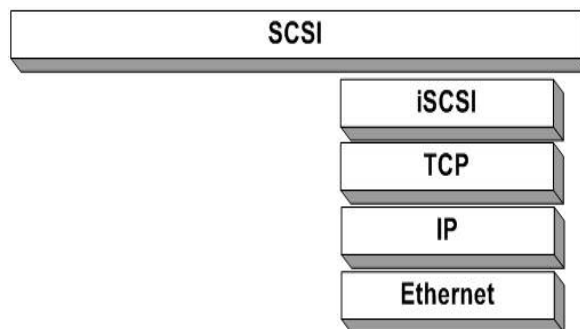


Figura 8. Protocolos de Transporte em Redes de Armazenamento. FONSECA (2008)

Uma das principais vantagens da utilização da tecnologia *iSCSI* é destacada por FONSECA (2008) como sendo “a possibilidade de utilização de redes existentes, sem a necessidade da implementação de uma rede dedicada baseada em componentes *fibre channel*”, cujo custo seria bem mais elevado.

Abaixo, a figura 9 apresenta o diagrama da estrutura *Oracle RAC 10g* através da utilização do recurso de *iSCSI*. A figura apresenta o exemplo de um *RAC* com 2 nós.

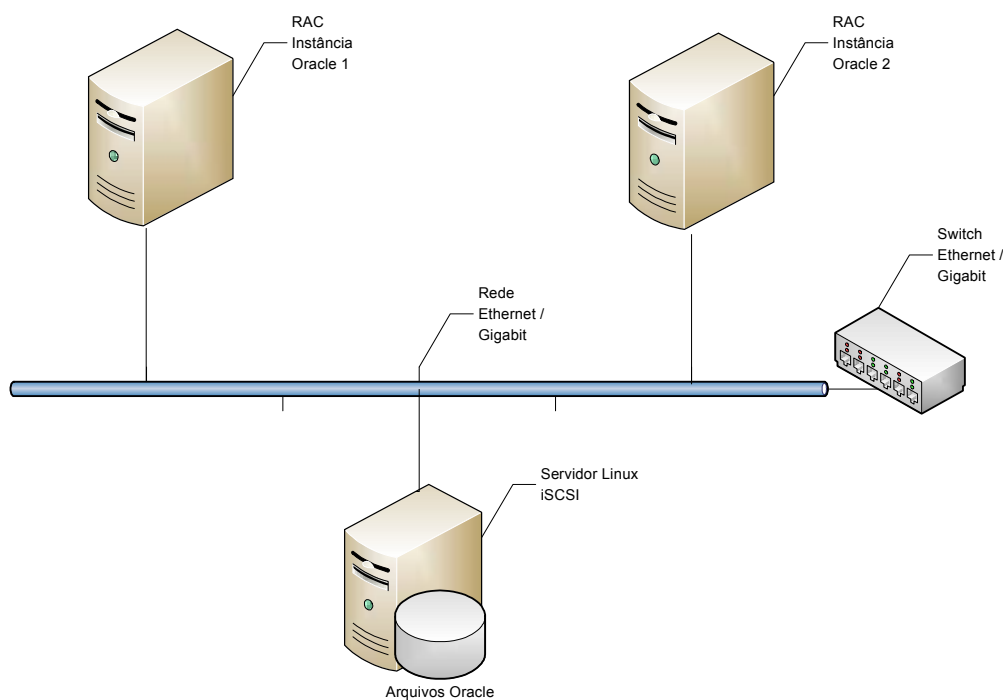


Figura 9. Diagrama da arquitetura Oracle RAC 10g através de *iSCSI*

4 CENÁRIO DE TESTES

O problema a ser pesquisado no projeto, se baseia em encontrar alternativas para a implementação de *RAC 10g* em ambientes nos quais não possuam em sua estrutura interna de TI um elemento indispensável para a implementação de tal solução, as chamadas *SAN's*, que nada mais são do que áreas de armazenamento de dados disponibilizados em rede.

A utilização de *SAN* em uma estrutura interna de rede propicia aos administradores de TI a simplificação da administração dos dados, conseguindo com isto uma flexibilidade muito grande, visto que estes espaços reservados ao armazenamento de dados em rede se encontram-se centralizados em um único lugar.

Porém, o problema central para a implementação de soluções *Oracle RAC 10g* é justamente o alto valor de mercado para a aquisição destes *storages*, que gira atualmente na casa de algumas dezenas de milhares de reais, inviabilizando com isto, o uso desta tecnologia para empresas de pequeno a médio porte, bem como instituições de ensino superior que muitas vezes sem recursos financeiros, não conseguem atualizar seus alunos que cursam disciplinas relacionadas a banco de dados com as tecnologias que estão sendo atualmente utilizadas no mercado de trabalho, ficando assim, estes alunos defasados em relação ao que se passa fora do mundo acadêmico.

Pensando neste problema, resolveu-se encontrar alternativas para a implementação de RAC sem que houvesse a necessidade da aquisição de *storages*, encontrando para isto, duas soluções que se encaixam perfeitamente para ambientes de testes e desenvolvimento das organizações, uma utilizando como solução a virtualização e outra utilizando a solução de *iSCSI* em sua estrutura de rede, sendo este recurso uma alternativa ao protocolo *SAN* por meio de um conjunto de comandos *SCSI* sobre o protocolo de rede *TCP/IP* disponível em redes *ethernet/gigabit* já disponíveis no ambiente das organizações. Finalizando, serão apresentados os testes de desempenho dentre as alternativas de implementação de *Oracle RAC 10g* sem a utilização de *storages*, facilitando com isso, a decisão sobre qual alternativa melhor se enquadraria ao ambiente em questão.

- Hipótese 1: Para a adoção de uma solução *RAC 10g*, seria necessária a aquisição de *storage* caso já não houvesse algum disponível, condição essencial para a implementação do sistema. Esta solução é a ideal para que se possa ter um desempenho esperado ao que a solução se propõe. Havendo tal recurso, não haveria a necessidade de utilização de sistemas virtualizados, nem mesmo a utilização de recursos de rede *iSCSI*, sendo somente necessária sua implementação e feitos os testes para confirmação de desempenho em relação á bases de dados hospedadas em servidores únicos.
- Hipótese 2: Não havendo a possibilidade de contar com o recurso de um *storage*, partiríamos para a implementação da solução através da utilização do recurso de

sistema virtualizado, fazendo com que o sistema “emule” os discos como se fossem um verdadeiro *storage*, com isto conseguindo a instalação do produto *Oracle RAC 10g* com sucesso. Nesta hipótese, serão realizados os testes necessários para a verificação de desempenho da base de dados Oracle nesta solução.

- Hipótese 3: Não havendo a possibilidade de contar com o recurso de um *storage*, partiríamos para a implementação da solução através da utilização do recurso *iSCSI*, no qual consiste em um servidor Linux fazendo o papel de *storage* através do protocolo *iSCSI* sobre *TCP/IP*, utilizando-se de toda a infra-estrutura de rede já existente no ambiente, como switches e cabeamento, fazendo com que os nós do cluster Oracle visualizem este servidor Linux como um *storage*, permitindo a implementação do produto *Oracle RAC 10g* com sucesso. Nesta hipótese, serão realizados os testes necessários para a verificação de desempenho da base de dados Oracle nesta solução.

Como objetivo geral, o projeto se propõe a esclarecer empresas e profissionais na área de TI que utilizem ou venham a trabalhar com bases de dados *Oracle* e optem por racionalizar seus recursos tecnológicos em nível de processamento, alta disponibilidade, padronização e automatização de seus recursos existentes, mas que não possuam o recurso necessário para tal implementação, como o caso de um *storage* em sua estrutura por exemplo, condição necessária e indispensável para a implementação da solução *Oracle RAC 10g*. O objetivo geral do projeto direciona para a viabilidade técnica e de desempenho sobre a possibilidade de uma solução em *grid* estar hospedada em sistemas virtualizados ou em soluções onde se adote a tecnologia de rede *iSCSI*, comparando o desempenho das bases de dados Oracle entre as duas alternativas propostas.

Como objetivo específico, o projeto se propõe a documentar as fases de instalação e configuração das ferramentas *Oracle RAC 10g*, bem como realizar testes comparativos de tempos de respostas a consultas entre uma base de dados *Oracle 10g* hospedada em um servidor virtualizado e outra base de dados com mesma estrutura, mas hospedada em um servidor que utilize o *iSCSI* em sua estrutura de rede.

O projeto em pauta propiciará às organizações uma nova visão em relação à possibilidade de gerenciamento de seus recursos de TI, resultando em um melhor aproveitamento dos recursos tecnológicos e conseqüentemente alcançando uma considerável

redução de custos em investimentos em TI, visto que a aquisição de supercomputadores responsáveis pelo processamento de dados possuem um custo extremamente elevado se tornando impeditivos para a maioria das empresas, uma vez levado este aspecto em conta, o projeto de clusterização sobre bases de dados *Oracle* visa utilizar recursos de *hardware* já disponíveis dentro da própria organização, alcançando juntas um poder de processamento equivalente ao de um supercomputador, muitas vezes superando-os em rapidez de processamento, sendo conseqüentemente mais eficientes.

Com isto, poderá vir a ser viável a utilização de alguma das alternativas abordadas para a implementação de *RAC 10g* dentro da própria Universidade Luterana do Brasil – Campus Guaíba – RS, especialmente em disciplinas de bancos de dados I e banco de dados II ministradas, propiciando aos alunos uma nova visão das tecnologias que hoje se apresentam no mercado, bem como demonstrar as vantagens da clusterização em banco de dados, no que tange á alta disponibilidade e desempenho fornecidas pela computação em *grid*.

Devido a possibilidade de empresas não contarem com *storages* ser alta, o que inviabilizaria a implementação de um *RAC 10g*, o projeto buscou duas alternativas para a implementação desta solução, uma utilizando sistemas virtuais e outra utilizando recursos de rede *iSCSI*. Com este propósito, o projeto visa identificar a viabilidade de utilização de tais recursos, através de comparativos de desempenhos das bases de dados *Oracle* entre as duas alternativas apresentadas, propiciando material de apoio para a análise sobre qual a solução que melhor se adaptaria ao contexto desejado.

4.1 Requisitos de *Hardware*

Neste capítulo será descrito qual seria o melhor cenário em nível de *hardware* para a implementação das duas soluções propostas na pesquisa (virtualização e *iSCSI*).

4.1.1 Virtualização

Conforme destaca ORACLE-BASE (2008), utilizando a ferramenta *VMware Server* pode-se executar várias máquinas virtuais (*VMS*) em um único servidor, permitindo que se

execute ambos nós do *RAC* em uma única máquina física. Além, permite-lhe criar discos virtuais compartilhados, superando o caro obstáculo do armazenamento compartilhado.

Porém, ORACLE-BASE (2008) nos alerta para as seguintes considerações a respeito do pré-requisitos de *hardware*:

- O sistema virtual proposto nesta pesquisa incluirá um sistema operacional do servidor hospedeiro, dois sistemas operacionais das máquinas virtuais, dois aplicativos *Oracle Clusterware*, duas instâncias *ASM* e duas instâncias *Oracle Database* todos localizados em um único servidor. Como pode-se imaginar, isso requer uma quantidade significativa de espaço em disco, CPU e memória. Testes realizados anteriormente com a instalação desta solução em uma máquina Pentium IV 3.4G com 2GB de memória *RAM*, falhou espantosamente. Quando testada em um servidor Intel Xeon 3.0G dual com 4GB de memória *RAM*, se comportou bem, mas a resposta não foi exatamente rápida.

Conclui-se com este dados, que o recurso da virtualização requer um servidor potente para que se consiga um desempenho razoável.

4.1.2 *iSCSI*

O recurso da utilização da estrutura de *iSCSI* para a implementação da solução *Oracle RAC 10g*, irá necessitar os seguintes *hardwares*:

- Um servidor Pentium IV 3.0 GHz com 2GB de memória *RAM* para a instância 01 do *RAC*;
- Um servidor Pentium IV 3.0 GHz com 2GB de memória *RAM* para a instância 02 do *RAC*;
- Um servidor Pentium IV 3.0 GHz com 1GB de memória *RAM* para instalação do pacote *iSCSI* para Ubuntu Server Linux 8.04.
- Estrutura de rede *ethernet/gigabit* existente.

4.2 Requisitos de Software

Neste capítulo estarão descritos os *softwares* necessários para a implementação das duas soluções propostas na pesquisa.

4.2.1 Virtualização

Para a implementação de *RAC 10g* sobre plataformas virtualizadas, será necessária a utilização dos seguintes *softwares*:

- *Ubuntu Server Linux 8.04 32 Bits (servidor hospedeiro)*;
- *VMware-server 1.0.6 (servidor hospedeiro)*;
- *Oracle Enterprise Linux 4 update 5 (em 2 servidores Oracle virtuais)*;
- *Oracle Database 10g R2 10.2.0.1 (em 2 servidores virtuais)*.

4.2.2 iSCSI

Para a implementação de *RAC 10g* sobre a arquitetura de *iSCSI*, será necessária a utilização de um sistema operacional Linux cujo *kernel* seja igual ou superior ao 2.6.16, além do seguinte *software*:

- *Ubuntu Server Linux 8.04 32 Bits (para 1 servidor iSCSI)*;
- *open-iscsi*;
- *Oracle Enterprise Linux 4 update 5 (em 2 servidores)*;
- *Oracle Database 10g R2 10.2.0.1 (em 2 servidores)*.

5 CONJUNTO DE TESTES PROPOSTOS

O conjunto de testes propostos na pesquisa engloba o comparativo de desempenho quanto ao tempo de resposta à determinada tabela com um número de registros consideráveis, uma em ambiente virtualizado e outra em ambiente *iSCSI*. Com estes testes serão gerados gráficos estatísticos demonstrando a viabilidade de utilização de cada alternativa proposta na pesquisa.

6 METODOLOGIA

A metodologia a ser adotada refere-se ao modo como serão realizados os testes de desempenho do banco de dados entre as soluções *iSCSI* e virtualizadas.

Atualmente, uma base de dados é projetada de maneira que se molde a qualquer situação possível, assim sendo de modo *standard*, nem sempre esta base de dados irá atingir o máximo de desempenho possível em determinadas ocasiões. Conforme destaca FLORES (2003), “a melhor maneira de melhorar o desempenho do banco de dados é realizando testes, que poderão verificar onde o banco se comporta de maneira mais eficaz e onde apresenta o pior desempenho”.

Um fator extremamente importante que afeta diretamente o desempenho de uma base de dados *Oracle*, são os parâmetros de inicialização do banco. A medida que sejam alterados certos parâmetros, deve-se monitorar o desempenho da base de dados quando são executadas consultas e atualizações para verificar o impacto das alterações.

A base de dados *Oracle* possui atualmente aproximadamente 250 parâmetros que configuram o sistema. Um administrador de banco de dados (DBA), é o responsável por fazer com que o banco de dados obtenha o máximo em relação ao desempenho, através da modificação dos valores de cada parâmetro. Alguns deles exigem a reinicialização do banco de dados, enquanto para outros, isto não se faz necessário. Conforme FLORES (2003), “um dos motivos pelo qual o *Oracle* tornou-se um poderoso e complexo SGBD, é o fato de possuir a habilidade de mudar dinamicamente sua configuração em tempo de execução”.

Para modificarmos os parâmetros de inicialização do *Oracle* pode-se utilizar comandos SQL que alterem os parâmetros textualmente, ou utilizar o pacote de ferramentas da Oracle chamado *Oracle Enterprise Manager (OEM)*.

Segundo LENKE (2002), existem três tipos gerais de parâmetros, que são caracterizados de acordo com o tipo de interrupções necessárias para mudar seu valor:

- **Parâmetros Estáticos:** É um pequeno subconjunto de parâmetros que não pode ser modificado sem que a base de dados seja completamente reinicializado. Um exemplo deste tipo de parâmetro é o `DB_BLOCK_SIZE`. O tamanho do bloco da base de dados é estático, pois as tabelas e os índices não podem ser movidos com diferentes tamanhos de blocos. Sua alteração exige uma recriação completa da base de dados;

- **Parâmetros Semi-Dinâmicos:** Não exigem que a base de dados seja totalmente reinicializada, mas as mudanças exigem que a base de dados permaneça indisponível por algum período de tempo;
- **Parâmetros Dinâmicos:** Esta classe de parâmetro permite alterar o tamanho e a configuração do sistema, com ele em funcionamento pleno;

Conforme FLORES (2003), estes parâmetros atuam em duas diferentes áreas: os parâmetros de *SGA* e os parâmetros de processo.

- **Parâmetros do SGA:** Quando um destes parâmetros é alterado, o *Oracle* dinamicamente reconfigura as regiões da memória, fazendo umas áreas menores e tornando outras maiores.
- **Parâmetros de Processo:** Afetam o comportamento dos processos internos. Por exemplo, é possível mudar o número de processos paralelos quantas vezes for desejado, pois o *Oracle* cria e destrói automaticamente processos internos sem afetar sua disponibilidade.

Para a avaliação de desempenho de suas bases, a *Oracle* desenvolveu alguns aplicativos específicos. Entre estes aplicativos podemos citar o *SQL TRACE*, o *TKPROF*, *Oracle Enterprise Manager (OEM)*, e o comando *timing*.

Além destas ferramentas citadas acima, será analisado o desempenho das bases através da ferramenta *Toad for Oracle* versão 9.0.1, o qual disponibiliza gráficos de desempenho em tempo real.

Os aplicativos descritos acima foram os escolhidos para a avaliação de desempenho das bases de dados clusterizadas nesta pesquisa.

A ferramenta *SQL TRACE* é uma ferramenta que fornece informações de desempenho de um comando *SQL* individual. O *SQL TRACE* pode ser programado para atuar sobre uma seção específica, coletando todas as informações e estatísticas de desempenho executadas na seção do usuário, arquivando-as em arquivos específicos.

Porém, FLORES (2003) destaca que estas informações contidas nestes arquivos são ilegíveis, de modo que é necessário usar a ferramenta de *TKPROF* para formatar estas informações e gerar uma saída de dados legível.

Segundo FLORES (2003), o *TKPROF* reporta cada sentença executada com cada pesquisa que foi usada. Também reporta o número de vezes que a pesquisa foi chamada e o

número de linhas que foram processadas. Assim, é possível localizar facilmente as sentenças que estão utilizando os maiores recursos. Portanto, uma ferramenta complementa a outra.

Primeiramente, destaca FLORES (2003), deve-se habilitar o serviço de SQL TRACE na base de dados, e logo após executar o comando *SQL*. Para ativar o serviço do SQL TRACE, é necessário o seguinte comando:

```
alter session set SQL_TRACE=true;
```

Um aspecto necessário a ser destacado é o fato da ferramenta *SQL TRACE* gerar uma sobrecarga demasiada no *SGBD*, sendo preciso desabilitar o serviço ao término das consultas que se deseja monitorar, apenas modificando o comando mencionado acima para *false*.

Embora a utilização de ferramentas de *TRACE* de sessões possam fornecer informações importantes e precisas sobre o desempenho de um banco de dados, até mesmo os pessoas da área de banco de dados podem ficar confusos com os resultados expressos pelos números, pois o volume de dados analisados tornam-se muito grandes e geralmente estes dados são confusos.

Já o comando *timing*, será muito útil, pois retornará o tempo exato de execução de cada consulta *SQL*.

Para ativá-lo, basta digitarmos o seguinte comando:

```
set timing on;
```

Já para desativá-lo, basta digitarmos o comando abaixo:

```
set timing off;
```

Uma alternativa para o problema em questão é a representação gráfica dos dados analisados, uma vez que gráficos são muito mais fáceis de entender e visualizar do que uma simples tabela com representações numéricas.

Pensando neste aspecto importante, a *Oracle* desenvolveu pacotes próprios que monitoram graficamente o desempenho do banco de dados. Um destes pacotes foi denominado de *Oracle Enterprise Manager (OEM)*.

Segundo FLORES (2003), “um dos aplicativos existentes no pacote é o *Oracle Performance Manager*, que permite monitorar o desempenho do banco, suas aplicações

relacionadas e sistemas operacionais em tempo real. Ele permite ao sistema e ao administrador do banco de dados monitorar as estatísticas, gravá-las e executá-las novamente mais tarde”.

Estas estatísticas gráficas podem ser demonstradas de várias formas através de tabelas, barras e gráficos disponibilizados dentro do aplicativo *OEM* acessado através de um *browser* de internet.

Abaixo, a figura 10, nos apresenta a interface web da ferramenta de administração *Oracle Enterprise Manager*.

The screenshot displays the Oracle Enterprise Manager (OEM) web interface for a Cluster Database named 'hdrac'. The browser window title is 'Oracle Enterprise Manager (SYSTEM) - Cluster Database: hdrac - Microsoft Internet Explorer'. The address bar shows the URL: 'http://es218201.fewiconsle/hdrac?st=op?type=rac_database&target=hdrac&event=dlload'. The interface includes a navigation menu with 'Home', 'Performance', 'Administration', and 'Maintenance'. The main content area is divided into several sections:

- General:** Status is 'Up', Up Instances: 2/2, Availability (%): 100.0, Cluster: 'sz_cluster', Database Name: 'HDRAC', Version: '10.1.0.1.0', Oracle Home: 'rdsle/emtest_hdrac/oracle'. A 'Shutdown' button is visible.
- Space Usage:** Problem: Tablespace: 0.
- High Availability:** Last Backup: Aug 11, 2003 2:08:55 AM, Archiving: Enabled, Flashback Logging.
- Advice:** Configuration: 3.
- Alerts:** A table showing the latest alerts. The table has columns for Severity, Target Name, Target Type, Category, Name, Message, Alert Triggered, Last Value, and Time.

Severity	Target Name	Target Type	Category	Name	Message	Alert Triggered	Last Value	Time
Warning	hdrac_hdrac1 Database	Database	Service CPU	Time (per user call)	The value of cpu_us for hdrac_hdrac1 is 41866	Aug 11, 2003 11:07:32 AM	70000	Aug 11, 2003 11:27:33 AM
Warning	hdrac_hdrac1 Database	Database	Dump Area	Dump Area Used (%)	98% of background dump area is used.	Aug 10, 2003 10:51:16 PM	97	Aug 11, 2003 11:26:48 AM
Warning	hdrac_hdrac1 Database	Database	Dump Area	Dump Area Used (%)	96% of core dump area is used.	Aug 10, 2003	97	Aug 11, 2003

Figura 10. Oracle Enterprise Manager

Conforme salienta BEST (2005), quando é instalado um banco de dados *Oracle*, também é instalado o *Oracle Enterprise Manager*. Está baseado em um controle via web da base de dados, servindo como o principal instrumento para administrar a base de dados *Oracle*. A ferramenta proporciona uma interface gráfica para fazer praticamente qualquer tarefa necessária a um banco de dados. Verificação gráfica de alertas e desempenho, criação e modificação de objetos e realização de rotinas de backup e recuperação são algumas das coisas que você pode fazer com o *Enterprise Manager*.

7 CONCLUSÕES

Independente do resultado das pesquisas de desempenho futuras a serem realizadas no TCC II, fica claro que a pesquisa tem por objetivo simplesmente oferecer alternativas para a implementação da computação em *grid* através da ferramenta de clusterização disponibilizada na versão 10g do banco de dados *Oracle*. Com os resultados de desempenho a serem conhecidos no TCC II, somente teremos uma noção de qual alternativa seja mais adequada ao propósito que se destina, ou seja, propiciar um ambiente alternativo e com um desempenho razoável ao uso de *storages* para armazenamento de dados em ambientes *Oracle RAC 10g*.

Através das pesquisas feitas no TCC I, ficou claro que empresas que possuam condições financeiras de utilizarem um excelente banco de dados como o Oracle – o melhor banco de dados disponível atualmente no mercado – e portanto, com valores de licenciamento bem significativos, teoricamente não teriam problemas significativos quanto á aquisição de unidades de armazenamento em rede – *storages* – para que a solução em RAC maximizasse ao extremo suas funcionalidades e seu desempenho.

Para o TCC-II, será realizada a etapa de implementação propriamente dita da pesquisa, englobando instalações, configurações, documentações e testes comparativos de desempenho entre as alternativas propostas na pesquisa.

8 REFERÊNCIAS BIBLIOGRÁFICAS

ALAPATI, Sam R. **Expert Oracle Database 10g Administration**. Berkeley, California, USA, 2005. 1305p.

ASHDOWN, Lance and LOGAN, Anna. **Oracle8i Standby Database Concepts and Administration**. Oracle Corporation, 1999. 188p.

BEST, Tom. **Oracle Database 10g: Administration Workshop I**. Oracle Corporation, 2005.

BURLESON, Consulting. **Oracle Concepts – SGA System Global Area**. Disponível em http://www.dba-oracle.com/concepts/sga_system_global_area.htm, Acesso em: 05 jun. 2008

CRIARWEB, **Criação de database links**. Disponível em

<http://www.criarweb.com/artigos/618.php>, Acesso em: 20 abr. 2008

DYKE, Julian and SHAW, Steve. **Pro Oracle Database 10g RAC on linux – Installation, Administration, and Performance**. Berkeley, California, USA, 2006. 807p.

FARIA, Elizabeth. **Grid e Oracle 10g**. Oracle Corporation, 2005.

FLORES, Felipe M. **Análise de Desempenho do Banco de Dados Oracle9i**. Canoas:ULBRA, 2003. (Trabalho de Conclusão de Curso)

FONSECA, Nelson e NETO, Antônio. **Fonseca and Neto: A Comparative Study Of The Performance**. Disponível em

<http://ieeexplore.ieee.org/iel5/9907/4378494/04378498.pdf?isnumber=4378494&prod=JNL&arnumber=4378498&arSt=150&ared=156&arAuthor=Fonseca%2C+Nelson+L.+S.+da%3B+Neto%2C+Antonio+J.+R>, Acesso em: 15 mai. 2008

HOWTOFORGE. **Setting up na iSCSI envirnoment on Linux**. Disponível em

http://howtoforge.com/iscsi_on_linux, Acesso em: 15 jun. 2008

LEMKE, Vilson Z. **Análise Comparativa dos Parâmetros Envolvidos no Ajuste de Desempenho dos Bancos de Dados Oracle9i e Interbase 6.5**. Gravataí:ULBRA, 2002. (Trabalho de Conclusão de Curso)

LONEY, Kevin. **Oracle Database 10g: The complete reference**. Emeryville, California, USA, 2004. 1348p.

ORACLE, **Oracle Press Release**. Disponível em

http://www.oracle.com/global/br/corporate/press/2008_jan/dados_embutidos.html, Acesso em: 10 mai. 2008

ORACLE-BASE, **Oracle 10g RAC on Linux using VMware Server**. Disponível em

<http://www.oraclebase.com/articles/10g/OracleDB10gR2RACInstallationOnCentos4UsingVMware.php>, Acesso em: 05 jun. 2008

RUSSO, Gilberto. **ISCSI - Um sistema de transferência de arquivos diferente**. Disponível em <http://www.vivaolinux.com.br/artigos/verArtigo.php?codigo=5809>, Acesso em: 15 mai. 2008

SHIELDS, Greg. **The shortcut guide to Selecting the right virtualization solution**, 2008. Realtime Publishers, 78p.

SYNGRESS, **Virtualization with VMware ESX Server**. Disponível em <http://www.syngress.com/catalog/?pid=3310>, Acesso em: 19 mai. 2008

THOME, Bob. **The High Availability Database Server Cookbook**. Oracle Corporation, 1999.

TO, Lawrence. **Oracle8i Standby Database**. Oracle Corporation, 1999.