

UNIVERSIDADE DO SAGRADO CORAÇÃO

THAIS MAIARA APARECIDA LEITE

**IMPLEMENTAÇÃO DE UM SISTEMA DE ARQUIVOS
DISTRIBUÍDOS: UM CASO PRÁTICO UTILIZANDO O
DFS**

BAURU
2013

THAIS MAIARA APARECIDA LEITE

**IMPLEMENTAÇÃO DE UM SISTEMA DE ARQUIVOS
DISTRIBUÍDOS: UM CASO PRÁTICO UTILIZANDO O
DFS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação do Prof. Esp. Henrique Pachioni Martins.

BAURU
2013

Leite, Thais Maiara Aparecida

L5334i

Implementação de um sistema de arquivos distribuídos /
Thais Maiara Aparecida Leite -- 2013.
65f. : il.

Orientador: Prof. Esp. Henrique Pachioni Martins.

Trabalho de Conclusão de Curso (Graduação em
Ciência da Computação) – Universidade do Sagrado
Coração – Bauru – SP.

1. Armazenamento distribuído. 2. Sistema de arquivos
distribuídos. 3. Distributed file system. I. Martins, Henrique
Pachioni. II. Título.

THAIS MAIARA APARECIDA LEITE

**IMPLEMENTAÇÃO DE UM SISTEMA DE ARQUIVOS
DISTRIBUÍDOS: UM CASO PRÁTICO UTILIZANDO O DFS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação do Prof. Esp. Henrique Pachioni Martins.

Banca examinadora:

Prof. Esp. Henrique Pachioni Martins
Universidade Sagrado Coração

Prof. Ms. Marcio Henrique Castilho Cardim
Universidade Sagrado Coração

Prof. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

Bauru, 03 de dezembro de 2013.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela vida que me deu e as oportunidades que me tem concedido. Agradeço aos meus pais, que mesmo enfrentando tanta dificuldade nessa vida sempre fizeram de tudo para que não faltasse nada a mim e ao meu irmão, financiando meus estudos, dando todo amor e apoio desde sempre. Ao meu orientador Professor Especialista Henrique Martins, por sua ajuda e disponibilidade para que as dúvidas fossem sempre esclarecidas. Aos meus amigos e em especial a Natália e o Nicolas, que contribuíram em pontos específicos do trabalho, como revisões, dúvidas. Agradeço a todos pela compreensão pela minha ausência.

RESUMO

Um sistema de armazenamento distribuído é uma abstração de acesso aos discos rígidos de microcomputadores dispersos fisicamente pela rede. Uma rede de armazenamento distribuído é composta de clientes e servidores, onde as estações podem desempenhar ambos os papéis. Sua implementação traz como obstáculo os principais desafios de sistemas distribuídos, como a transferência, a replicação, a segurança, a concorrência, entre outros. Desta forma este trabalho teve como objetivo à implementação de um sistema de arquivos distribuídos (SAD), onde foi configurado o *Distributed File System* (DFS) em servidores virtuais para que o usuário possa armazenar e manipular arquivos sem que saiba a localização física dos mesmos, facilitando o processo de armazenamento e compartilhamento de arquivos com segurança. O estudo apresenta fundamentos sobre sistemas de arquivos distribuídos, citando alguns dos sistemas que mais se utilizam e exemplificando um cenário no DFS, isto define o trabalho de implementação de um sistema de arquivo distribuído. Os resultados deste estudo comprovam o quão eficiente pode ser um SAD dentro de uma organização.

Palavras Chave: Armazenamento Distribuído. Sistemas de Arquivos Distribuídos.

Distributed File System.

ABSTRACT

A distributed storage system is an abstraction of access to the hard drives of computers physically dispersed over the network. A distributed storage network consists of clients and servers, where stations can play both roles. Its implementation has as the main obstacle challenges of distributed systems, such as transfer, replication, security, competition, among others. Therefore, this study aimed to implement a distributed file system, which was set up the Distributed File System (DFS) on virtual servers so that the user can store and manipulate files without knowing the physical location of customers, facilitating the process storage and share files safely. The study presents the fundamentals of distributed file systems, citing some of the systems that are used more and illustrating a scenario in DFS, this defines the job of implementing a distributed file system. The results of this study show how efficient can be a DFS within an organization.

Key-words: Distributed Storage. Distributed File System. Distributed File System

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura AFS.....	35
Figura 2 - Distribuição dos processos no Andrew File System.....	37
Figura 3 - Espaço de nomes vistos de um cliente AFS.	38
Figura 4 - Ambiente virtualizado.	43
Figura 5 - Máquinas virtuais dos servidores em execução.....	44
Figura 6 - Máquina virtual do cliente.....	44
Figura 7 - Configuração do servidor de arquivos.....	45
Figura 8 - Acesso ao diretório do domínio.	46
Figura 9 - Adição de um novo <i>Namespace</i>	47
Figura 10 - Configuração da replicação do DFS.....	48
Figura 11 - Mapeamento da unidade dentro do DFS.....	49
Figura 12 - Mapeamento criado.....	50
Figura 13 - Criação do arquivo na máquina cliente.	51
Figura 14 - Replicação do arquivo nos servidores.....	52

LISTA DE ABREVIATURAS E SIGLAS

ADDS - Active Directory Domain Services

AFS – Andrew File System

AVSG - Accessible Volume Storage Group

DFS - Distributed File System

FTP - File Transfer Protocol

ISO - International Organization for Standardization

NFS - Network File System

SAD - Sistema de Arquivos Distribuídos

TCP - Transfer Control Protocol

UDP - User Datagram Protocol

VFS - Virtual File System

VSG - Volume Storage Group

WAN - Wide Area Network

SUMÁRIO

1 INTRODUÇÃO	10
2 OBJETIVOS	12
2.1 OBJETIVO GERAL.....	12
2.2 OBJETIVO ESPECÍFICO	12
3 JUSTIFICATIVA	13
4 REVISÃO DA LITERATURA	15
4.1 SISTEMA DE ARQUIVOS	15
4.1.1 Nomes de arquivos.....	16
4.2 SISTEMA DE ARQUIVOS DISTRIBUÍDOS	18
4.2.1 Serviço de nomes.....	19
4.2.2 Serviço de arquivos	20
4.2.3 Serviço de diretórios.....	20
4.3 CARACTERÍSTICAS DESEJÁVEIS EM SISTEMAS DE ARQUIVOS DISTRIBUÍDOS	21
4.3.1 Atualização concorrente	21
4.3.2 Replicação.....	22
4.3.3 Heterogeneidade.....	24
4.3.4 Tolerância a falhas	24
4.3.5 Consistência.....	25
4.3.6 Segurança	25
4.3.7 Eficiência	29
4.3.8 Transparência	29
4.4 FERRAMENTAS DE SISTEMAS DE ARQUIVOS DISTRIBUÍDOS	31
4.4.1 Coda file system.....	31
4.4.2 Network file system - NFS	33
4.4.2.1 Arquitetura - Network File System	33
4.4.2.2 Cache – Network File System	35
4.4.2.3 Segurança -Nnetwork File System	36
4.4.3 Andrew File System - AFS	36
4.4.4 Distributed File System - DFS	39
4.4.5 Google File System	40

5 METODOLOGIA	42
6 RESULTADOS	43
6.1 IMPLEMENTAÇÃO DO DFS.....	43
6.2 TRANSPARÊNCIA DE ACESSO	48
6.3 TRANSPARÊNCIA DE REPLICAÇÃO E LOCALIZAÇÃO.....	50
7 CONSIDERAÇÕES FINAIS	53
REFERÊNCIAS	54

1 INTRODUÇÃO

No começo da década de 70, foram desenvolvidos os disquetes, e devido ao pequeno tamanho dos arquivos era considerado um grande dispositivo de armazenagem e com o passar dos anos foram surgindo novas formas de compartilhar e armazenar arquivos como CDs, DVDs, cartão de memória, Pen Drive, Blu-Ray, além da existência de armazenamento de arquivos, tais como armazenamento distribuído.

Segundo Tanenbaum (2008) compartilhar dados é fundamental para sistemas distribuídos sejam a base para muitas aplicações distribuídas, de um modo seguro e confiável. De acordo com Coulouris, os sistemas de arquivos distribuídos básicos fornecem um apoio fundamental para computação organizacional baseada em intranets, possibilitando que os usuários acessem seus arquivos a partir de qualquer computador (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Um sistema distribuído consiste de computadores independentes ligados uns com os outros. A diferença básica entre um sistema distribuído e uma rede de computadores é que em um ambiente distribuído os recursos são disponibilizados para o usuário de forma transparente. Isso significa que, teoricamente, os usuários nem sabem que os computadores estão interconectados. (GALLO. 2003)

Por sua vez, Silbershatz (2008) define um sistema de arquivos distribuído (*Distributed File System - DFS*) como uma implementação distribuída do modelo clássico de tempo compartilhado de um sistema de arquivos, no qual vários usuários compartilham arquivos e recursos de armazenamento (SILBERSHATZ,2008).

A utilização de uma DFS é feita em grandes corporações, cuja rede de computadores é uma WAN (Wide Area Network), possibilitando assim o gerenciamento dos arquivos compartilhados e replicando arquivos em outro servidor

para que haja disponibilidade. Hoje usuários podem criar, acessar, modificar seus arquivos sem se preocupar onde está localizado, pois acessam os arquivos com transparência, segurança e eficiência.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Mostrar o funcionamento de arquivos distribuídos no ambiente Microsoft *Windows Server 2008 Enterprise R2*.

2.2 OBJETIVO ESPECÍFICO

- Mostrar a praticidade da implementação do DFS.
- Mostrar o funcionamento do DFS.
- Desenvolver testes como transparência de acesso, de replicação e localização na plataforma Microsoft *Windows Server 2008 Enterprise R2*.

3 JUSTIFICATIVA

O grande aumento do volume de dados exige maiores condições de armazenamento e processamento compatível. E por mais que se tenham estes dados centralizados em servidores ou estações simples, os mesmos estão sujeitos a falhas e sobrecargas, gerando interrupções no fornecimento do serviço.

Mesmo com a evolução dos *hardwares* e *softwares*, sempre há a possibilidade de acontecer algum incidente, como por exemplo, os discos rígidos falharem, interrupções da rede de dados, as sobrecargas nos servidores dentre outros, isso acabou nos levando a descentralização dos serviços com a finalidade de diminuir a perda total dos recursos, portanto, as falhas podem ocorrer, porém na maioria dos casos afetarão somente uma porção dos serviços e não todo conjunto.

Apoiado sobre a ideia de compartilhar recursos, a distribuição na camada do sistema de arquivos para redes traz o conceito de Sistema de Arquivos Distribuído (SAD), ou seja, um sistema de arquivos no qual os arquivos estão armazenados e distribuídos em computadores diferentes interligados através de uma rede (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Dentro deste contexto, um SAD deverá ter uma solução eficiente para o gerenciamento de arquivos distribuídos e, na medida do possível possa apresentar características como transparência de acesso e localização, escalabilidade, desempenho, tolerância a falhas, controle de concorrência e segurança.

A interligação destes componentes agindo como um todo traz vantagens no uso mais eficiente dos ativos computacionais, a relação a um crescimento direcionado aos recursos, a disponibilidade mediante a replicação dos componentes e arquivos. Contudo, algumas desvantagens são presentes como, por exemplo, a incerteza da rede de comunicação, a segurança dos dados, a complexidade na

cooperação entre componentes e a dificuldade do gerenciamento dos recursos distribuídos.

Com base nisso, o presente trabalho visa a implantação do SAD que tem por finalidade de implementar um modelo de DFS que possa ajudar no gerenciamento dos arquivos distribuídos, com total transparência, desempenho, escalabilidade, controle de concorrência, tolerância a falhas e segurança que o usuário não perceberá os incidentes, caso ocorra.

4 REVISÃO DA LITERATURA

4.1 SISTEMA DE ARQUIVOS

O sistema de arquivos é uma camada importante do sistema operacional, pois provê uma abstração do armazenamento secundário (discos rígidos, discos magnéticos etc.) e é responsável pela nomeação global, o acesso aos arquivos e a organização destes (GALLI, 2000).

As operações realizadas em computador envolvem programas e aplicações. Os programas em execução e toda a estrutura responsável para manter informações necessárias à sua execução são chamados de processos.

Qualquer aplicação realizada através de computador precisa armazenar recuperar dados e muitas vezes compartilhá-los com outros processos. O modo como os usuários precisam lidar com estas tarefas depende de como o sistema operacional organiza e mantém estas informações nos meios de armazenamento, mediante a implementação de arquivos.

Os arquivos são unidades que contém informações logicamente relacionadas, podendo constituir-se de programas ou dados. Os programas contém instruções compreendidas pelo computador, e os arquivos de dados podem conter qualquer tipo de informações, como textos, registros de um banco de dados, etc.

A parte do sistema operacional responsável pelo gerenciamento e manutenção dos arquivos é denominada de sistema de arquivos. Através do sistema de arquivos, tarefas de recuperação e acesso ao conteúdo dos arquivos são disponibilizadas ao usuário, que ficam dispensados dos detalhes de como estes arquivos estão organizados nos meios de armazenamento, como é o controle de memória disponível e outros detalhes da implementação. Para os usuários, o

aspecto a ser considerado é como um arquivo será identificado, como será protegido e quais as operações que podem ser realizadas sobre ele.

O sistema de arquivos controla a forma como os arquivos são organizados, como será o acesso aos dados, como serão identificados pelo usuário, quais os atributos e operações possíveis sobre eles.

É através do sistema de arquivos que os usuários terão uma interface para armazenar e recuperar seus dados, de forma transparente quanto aos detalhes de implementação e organização. E é através dele também que os diferentes processos do sistema poderão executar tarefas sobre os arquivos ou compartilhá-los com outros processos (TANENBAUM, 1995; MACHADO, 1997).

Os sistemas de arquivos são projetados para armazenar e gerenciar um grande número de arquivos, com recursos para criação, atribuição de nomes e exclusão de arquivos. Os sistemas de arquivos também assumem a responsabilidade pelo controle de acesso aos arquivos, restringindo o acesso de acordo com as autorizações dos usuários e com o tipo de acesso solicitado (leitura, atualização, execução etc.) (COULOURIS, 2005; TANENBAUM, 2000; SILBERSCHATZ, 2000).

O sistema de arquivos pode oferecer serviços para atender as necessidades de armazenamento, controle e recuperação de arquivos e também serviços para a organização desses arquivos em diretórios.

4.1.1 Nomes de arquivos

Nos sistemas de arquivos que utilizam estrutura de árvore de diretórios, existem regras para a formação dos nomes dos arquivos. Dois métodos são

utilizados para a formação de nomes de arquivos: nome de caminho absoluto e nome de caminho relativo.

No nome de caminho absoluto, o nome do arquivo é composto pelo caminho do diretório raiz até o arquivo. Por exemplo, /dir1/dir2/x significa que o diretório raiz contém um diretório chamado dir1, e que este contém um subdiretório chamado dir2, e que este por sua vez contém um arquivo chamado x.

No nome de caminho relativo, utiliza-se do conceito de diretório corrente ou diretório de trabalho. Neste método, o usuário designa um diretório para ser o diretório corrente, e todos os arquivos referenciados sem o diretório raiz no caminho, são considerados relativos ao diretório corrente. Por exemplo, se o diretório corrente é /dir1/dir2, então o arquivo cujo nome absoluto é /dir1/dir2/x pode ser referenciado apenas por x (TANEMBAUM, 1987).

Silberschatz (2000) enfatiza que “Idealmente, um serviço de sistema de arquivo distribuído deve aparecer aos seus clientes como um sistema de arquivos centralizado-convencional. A multiplicidade e a dispersão de seus servidores e dispositivos de armazenamento devem ser transparentes. Ou seja, a *interface* cliente desse serviço não deve fazer distinção entre arquivos locais e remotos. Cabe ao serviço localizar os arquivos e organizá-lo para o transporte dos dados. Um sistema de arquivos distribuído transparente facilita a mobilidade do usuário trazendo o ambiente do usuário (ou seja, seu diretório inicial) para onde quer que o usuário efetue login” (SILBERSCHATZ, 2000, p. 542). Diversas técnicas de transparência, adotadas em sistemas distribuídos são empregadas parcialmente ou diretamente na implementação de serviços de arquivos (COULOURIS, 2005; KON, 1994).

4.2 SISTEMA DE ARQUIVOS DISTRIBUÍDOS

Existem diversas maneiras diferentes de se expressar o que é um sistema distribuído, dentre elas podemos destacar três mais relevantes.

A definição de Lamport (1978) diz que: “Um Sistema Distribuído consiste de uma coleção de processos distintos que são especialmente separados e que se comunicam por troca de mensagens”.

Já em Coulouris (2005), é especificado como: “Um sistema distribuído é aquele no qual os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas por troca de mensagens”.

E por fim, Tanenbaum e Steen (2007) definem como: “Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente”.

A construção de sistemas distribuídos vem da motivação em compartilhar recursos computacionais de *hardware*, *software* e dados, o que envolve alguns problemas como concorrência, inexistência de um relógio global, falhas independentes, dentre outras mais (COULOURIS, 2005).

Vários conceitos importantes devem ser observados para a construção de um sistema distribuído, como a transparência na distribuição em seus diferentes níveis, fornecer acesso fácil aos recursos do sistema, o tratamento eficiente das falhas no sistema quando estas acontecerem, e permitir que o sistema possa ser expandido (TANENBAUM E STEEN, 2007).

Um Sistema de Arquivos Distribuídos (SAD) tem o objetivo de fornecer os mesmos serviços e recursos de um sistema de arquivos convencional, ao menos na ótica dos clientes que os utiliza, com a diferença que este pode ser acessado de qualquer máquina dentro de uma rede (acesso remoto).

Serão apresentados três serviços intrínsecos aos sistemas de arquivos distribuídos que facilitam o acesso às informações para os usuários, escondendo toda a complexidade existente na nomeação global, no acesso aos arquivos dispersos pela rede e na organização destes que são o serviço de nomes, o serviço de arquivos e o serviço de diretórios, respectivamente (GALLI, 2000).

4.2.1 Serviço de nomes

O serviço de nomes tem a função de fornecer a localização de um determinado arquivo no conjunto de computadores do sistema de arquivos distribuído. Caso o nome do arquivo contiver o nome do computador aonde ele está localizado, como por exemplo, "apolo:/tmp/arquivo", então este serviço não provê transparência de localização. Para prover esta transparência, o nome de um arquivo não deve possuir indícios de sua localização física, pois caso ele mude de lugar ou possua várias cópias, o seu nome ou caminho não precisarão ser alterados (GALLI, 2000).

Segundo Tanenbaum e Steen (2007), o serviço de nomes tem como finalidade indicar onde um determinado arquivo está localizado informando o seu nome ou caminho. Caso a localização do arquivo esteja armazenada no seu nome então esse serviço de nomes não provê transparência de localização. Para que se tenha transparência, o nome ou caminho de um arquivo não deve ter indícios de onde ele está fisicamente localizado. Caso esse arquivo seja movido para outra localização, ou que possua várias cópias, o seu nome ou caminho não precisará ser modificado.

4.2.2 Serviço de arquivos

É de responsabilidade do serviço de arquivos fornecerem operações sobre os recursos (arquivos) que compõe o sistema. O armazenamento dos arquivos pode ser realizado de formas distintas, levando em conta seu tipo e sua utilidade. Como exemplo, arquivos pertencentes a um banco de dados podem ter seu armazenamento em forma de registros. Arquivos utilizados em aplicações multimídia normalmente são armazenados em formato contínuo no disco, com a finalidade de obter uma melhor leitura, etc.

O serviço de arquivos distribuído tem como responsabilidade também as propriedades dos arquivos, as quais não são armazenadas nos mesmos, tais como a data de criação, data de alteração, tamanho, dono do arquivo, permissões de leitura, escrita e execução, além de outras informações relevantes a respeito do mesmo.

Existe a possibilidade também que esse serviço trabalhe com replicação de arquivos, o que facilita desta forma o acesso concorrente, propicia assim mais velocidade para as aplicações dos clientes, e ajuda na disponibilização dos arquivos, caso ocorra algum problema com um servidor ou até mesmo que ele fique fora do ar (COULOURIS; DOLLIMORE; KINDBERG, 2007).

4.2.3 Serviço de diretórios

O objetivo do serviço de diretórios é manter a organização dos arquivos armazenados no sistema, fornecendo ao usuário uma *interface* para que eles possam organizar seus arquivos em uma estrutura hierárquica, através de diretórios e subdiretórios (GALLI, 2000).

As principais operações nos diretórios fornecidas pelos serviços de diretórios são criação, remoção, alteração, listagem, alteração de permissões, etc. Além destas operações citadas, esse serviço também influencia no gerenciamento de arquivos, em operações de criação, remoção, renomeação, busca, duplicação, entre outras.

4.3 CARACTERÍSTICAS DESEJÁVEIS EM SISTEMAS DE ARQUIVOS DISTRIBUÍDOS

Muitos dos conceitos encontrados na implementação de sistemas distribuídos devem ser levados em consideração no desenvolvimento de sistemas de arquivos distribuídos. Inicialmente, os primeiros sistemas de arquivos distribuídos ofereciam recursos de transparência de acesso e transparência de localização, emergindo subseqüentemente à preocupação no desenvolvimento de recursos como desempenho, escalabilidade, controle de concorrência, tolerância a falhas e segurança (COULOURIS, 2005).

4.3.1 Atualização concorrente

A concorrência segundo Coulouris, Dollimore e Kindberg (2007), se caracteriza quando usuários podem acessar vários arquivos, ou os mesmos arquivos de forma simultânea, sem sofrer danos, perda de desempenho ou outras restrições. Estes acessos devem ser realizados sem que o usuário necessite saber como é realizado pelos servidores. Dessa forma, salienta-se que a transparência de concorrência e de paralelismo é uma importante característica dos Arquivos Distribuídos.

O problema principal na solução deste tipo de implementação é em relação à sincronização dos arquivos, incluindo leitura e escrita concorrente. Implementar a leitura concorrente pode ser realizada de forma fácil, caso não exista a necessidade de escrita concorrente. Quando um arquivo estiver sendo realizada uma operação de leitura, não poderá ser aplicada uma operação de escrita.

À escrita concorrente apresenta uma complexidade maior, pois quando um cliente escreve num arquivo, todos os seus leitores devem ser notificados sobre esta alteração, e todos seus escritores precisam tomar cuidado para não escreverem em cima das modificações realizadas por outros clientes. Neste caso a última alteração é a que apresenta validade, ou os escritores discutem entre si para tentarem fazer uma agregação das alterações.

Em um ambiente distribuído, o problema é agravado, porque além da necessidade de se preocupar com operações de leitura e escrita de forma concorrente, é necessária que a carga seja distribuída entre os vários servidores e mantida a sincronização quando ocorrer alguma operação de escrita, da forma mais rápida (ou mais coerente) possível, mantendo assim os dados consistentes.

4.3.2 Replicação

Caso um sistema de arquivos tenha suporte à replicação, a confiança e a eficiência do serviço de arquivos são aumentadas de forma significativa. Eficiência em relação a tempo de resposta, carga do servidor e tráfego de rede, conceitos que podem ser verificados em Couloris, Dollimore e Kindberg (2007). Adicionalmente, se um servidor sair do ar, o serviço de arquivos ainda pode manter suas obrigações.

Desta maneira a replicação de arquivos provém tolerância a falhas, uma vez que o usuário não consegue perceber que o servidor o qual estava conectado caiu e

que outro (ou outros) entrou no seu lugar para fornecer o arquivo que ele estava utilizando. Assim é interessante que o sistema também ofereça transparência de replicação, pois não é necessário o usuário saber como o sistema trata da replicação deste arquivo.

Alguns sistemas de arquivos distribuídos oferecem o serviço de replicação de arquivo. Isso significa ter várias cópias de arquivos selecionados, que serão mantidos em servidores distintos.

A replicação de arquivos traz alguns benefícios:

Torna o sistema mais confiável, pelos diversos *backups* que possui. Se um dos servidores tiver problemas, o arquivo fica acessível através de outro servidor. Ou se ocorrer uma falha irreversível no sistema de arquivos de um servidor, não haverá perdas de dados;

É possível acessar um arquivo mesmo se um servidor tiver problemas;

A replicação pode trazer também ganhos de performance. Se um servidor estiver muito carregado, pode-se acessar um arquivo através de outro servidor que esteja mais liberado.

O principal desafio associado à replicação é a sua atualização e manutenção da consistência entre as réplicas dos arquivos nos diversos servidores. Para o usuário, as réplicas de um arquivo denotam a mesma entidade lógica e, dessa forma, uma atualização a qualquer réplica deve refletir nas demais. Os detalhes da replicação devem ser transparentes para os usuários, sendo tarefa do esquema de nomeação mapear um nome de arquivo replicado em um computador específico sem o conhecimento de sua localização pelo cliente.

Poucos serviços de arquivos suportam replicação completa, mas a maioria suporta o armazenamento de arquivos, ou de porções de arquivos em caches locais,

que é uma forma limitada de replicação (COULOURIS, 2005; SILBERSCHATZ, 2000; KON, 1994).

4.3.3 Heterogeneidade

Os SADs devem permitir acesso a um conjunto heterogêneo de computadores e redes, aplica-se à redes, *hardware* de computadores, sistemas operacionais, linguagens de programação, implementação de diferentes programadores.

Cada vez mais, buscam-se padronizações que permitam que máquinas de diferentes fabricantes se comuniquem sem grandes dificuldades. O sistema de arquivos distribuído deve ser implementado de forma que o servidor e o cliente não necessitem da mesma arquitetura de *hardware* e da mesma solução de *software* para a correta execução do serviço (COULOURIS, 2005; KON, 1994).

4.3.4 Tolerância a falhas

Por ser parte essencial nos sistemas distribuídos, é essencial que o serviço de arquivos distribuídos continue a funcionar diante de falhas de clientes e servidores e a principal regra de um serviço de arquivos em um ambiente de sistemas distribuídos é garantir o atendimento as requisições dos usuários caso algum servidor venha a falhar (COULOURIS, 2005, 2007).

A tolerância a falhas segundo Tanenbaum e Sten (2007) é que mesmo que um servidor fora do ar ou da rede, o sistema de arquivos deve manter as

informações e ficar disponível de forma total ou parcialmente. Além disso, os usuários não necessitam saber como isso foi implementado e nem como funciona.

O método mais utilizado para aumentar a disponibilidade de um serviço de arquivos tem sido a replicação de dados, onde os arquivos são armazenados em dois ou mais servidores e caso um deles não esteja disponível, outro servidor poderá fornecer os serviços solicitados (COULOURIS, 2005; KON, 1994).

4.3.5 Consistência

Um sistema de arquivos distribuídos deve garantir a consistência dos arquivos de seus usuários. Quando um arquivo é replicado ou recuperado do cache de uma estação cliente, por exemplo, podem ocorrer demoras inevitáveis na propagação das modificações devido a latências de rede, podendo resultar na geração de inconsistências ou até perda das informações caso o sistema seja frágil a esse tipo de cenário (COULOURIS, 2005; SILBERSCHATZ, 2000).

4.3.6 Segurança

A finalidade da camada de segurança em sistemas distribuídos é de restringir o acesso à informação e aos recursos para usuários e processos não autorizados, fortalecendo a privacidade de indivíduos e organizações. Muitos dos recursos existentes nos sistemas distribuídos lidam com informações de elevado valor intrínseco para os usuários (COULOURIS, 2005), o que justifica a importância da segurança nestes sistemas.

Implementar técnicas de segurança em sistemas distribuídos é mais complexo em relação aos sistemas centralizados devido a falta de um único ponto

de controle e do uso de redes inseguras para a transmissão de dados (SINHA, 1996).

Tanenbaum (2002) diz que a segurança em sistemas computacionais esta relacionada fortemente com a noção de confiança dos usuários, que usufruirão dos serviços. Os serviços de segurança caracterizam os diferentes aspectos de um sistema de computador, tais como (STALLINGS, 2003):

Autenticidade: Requer que a origem ou o originador de uma mensagem seja corretamente identificado. A verificação da autenticidade é necessária após todo processo de identificação, seja de um usuário para um sistema, de um sistema para o usuário ou de um sistema para outro sistema.

Integridade: Consiste em proteger a informação contra modificação sem a permissão explícita do proprietário daquele dado. A modificação inclui ações como escrita, alteração de conteúdo, alteração de *status*, remoção, criação e o atraso de informações transmitidas.

Confidencialidade: Consiste em proteger a informação contra leitura ou cópia por alguém que não tenha sido explicitamente autorizado pelo proprietário. Este tipo de segurança inclui não apenas a proteção da informação como um todo, mas também de fragmentos dela, que podem ser utilizadas para inferir sobre o todo.

Controle de Acesso: Refere-se a capacidade de se permitir ou negar acesso aos serviços e recursos oferecidos pelo sistema. Acessos desconhecidos ou feitos por pessoas não autorizadas podem significar a necessidade de uma verificação de todos os recursos envolvidos em busca de possíveis estragos que possam ter sido causados ao sistema, mesmo que aparentemente nada tenha ocorrido.

Para proteger os serviços e informações contidas nos sistemas computacionais, deve-se ficar atento às ameaças de segurança, como ataques de

negação de serviço, que podem gerar perdas ou danos à informação. De acordo com Tanenbaum (2002), há quatro tipos de ameaças de segurança que se deve considerar:

- *Intercepção*: Ocorre quando um componente do sistema é acessado sem autorização.
- *Interrupção*: Refere-se a situação em que o componente do sistema é corrompido ou torna-se indisponível, como por exemplo, um rompimento de uma linha de comunicação.
- *Modificação*: Envolve a alteração de um componente do sistema por parte de um usuário não autorizado, como por exemplo, a modificação do valor de pagamento de uma compra via Internet durante o trânsito das informações pela rede.
- *Fabricação*: Refere-se a situações onde são inseridos objetos falsos em um componente do sistema.

Com o intuito de evitar essas ameaças, Coulouris (2005) e Tanenbaum (2002) mencionam alguns mecanismos e técnicas para a segurança em sistemas distribuídos e aplicativos, tais como:

- *Criptografia*: A criptografia é o processo de codificar uma mensagem fazendo-se uso de algoritmos e chaves de segurança. A chave criptográfica é um parâmetro utilizado nos algoritmos de criptografia que garante que ela possa ser invertida sem o conhecimento da chave. Existem dois métodos de criptografia. No primeiro, denominado criptografia simétrica, o emissor e o receptor compartilham uma chave secreta que não pode ser de conhecimento de terceiros. Esta chave é utilizada para criptografar e descriptografar a mensagem. Na segunda

técnica, denominada de criptografia assimétrica, são utilizados pares de chaves para criptografar e decifrar as mensagens. O emissor possui uma chave pública do destinatário e este faz uso dela para criptografar a mensagem antes de enviá-la. O destinatário utiliza a chave privada correspondente para decifrar a mensagem.

- *Autenticação:* A autenticação é o mecanismo utilizado para garantir a real identidade de um usuário ou processo. A autenticação pode ser feita de forma unilateral, mútua ou com a mediação de terceiros, dependendo da situação. A autenticação unilateral acontece quando apenas um processo da comunicação autentica-se perante o outro, mas a recíproca não é verdadeira. Na autenticação mútua os processos envolvidos na comunicação se autenticam um perante o outro. A autenticação com a mediação de terceiros é utilizada quando os processos da comunicação não se conhecem e dessa forma não podem se autenticar mutuamente, mas conhecem um terceiro com quem se autenticam e recebem credenciais para procederem assim a autenticação mútua.
- *Controle de Acesso:* As técnicas de controle de acesso propõem-se a definir quais usuários ou processos possuirão acesso a um determinado recurso do sistema computacional e quais as permissões que a ele serão concedidas. Estas técnicas podem ser aplicadas em qualquer nível, desde a aplicação, definindo quais usuários possuirão acesso a determinados registros, até ao sistema básico, determinando quais processos possuirão acesso a determinadas páginas de memória.

4.3.7 Eficiência

Um sistema de arquivos distribuído deve fornecer um serviço que seja comparável (ou melhor) aos sistemas de arquivos locais, em termos de desempenho e confiabilidade. Para isso foram desenvolvidas técnicas para contornar os atrasos ocorridos com a comunicação entre os nodos, entre elas destacam-se o cache, que acessa a rede o mínimo possível, armazenando as informações acessadas constantemente na memória e também o balanceamento de carga, que graças a existência de replicas dos arquivos espalhadas em diversos servidores, pode optar por requisitar um arquivo de um ou mais servidores que estejam menos sobrecarregados, ganhando assim em tempo de processamento (COULOURIS, 2005).

4.3.8 Transparência

A transparência tem por objetivo esconder, sempre que possível, todos os detalhes do sistema que são irrelevantes do usuário (CHOW, 1998). Alcançar a transparência completa é uma tarefa complexa e requer que diversos aspectos, identificados pela *International Organization for Standardization* (ISO), sejam levados em consideração durante o desenvolvimento (COULOURIS, 2005; TANENBAUM, 2002):

Transparência de Acesso: Permite que o acesso a recursos locais e recursos remotos seja feito através das mesmas operações.

Transparência de Localização: Permite que os recursos possam ser acessados sem o conhecimento de sua real localização. A transparência de localização possibilita que recursos sejam movidos entre nodos de um sistema distribuído, como por exemplo, um arquivo, sem ter seu nome e caminho afetado.

Transparência de Concorrência: Permite que os diversos processos possam operar concorrentemente usando um recurso compartilhado sem que cause interferência aos demais processos que estão utilizando-o.

Transparência de Replicação: Permite que os usuários façam uso das múltiplas instancias dos recursos com o intuito de aumentar a confiabilidade e o desempenho sem que estes saibam que estão manuseando réplicas.

Transparência de Falha: Permite o encobrimento das falhas de *software* e *hardware*, possibilitando aos usuários a conclusão de suas tarefas sem que tomem conhecimento dos possíveis fracassos do sistema.

Transparência de Migração: Permite o movimento de recursos e clientes dentro de um sistema sem afetar a operação do usuário ou da aplicação e sem que estes tomem conhecimento desta operação.

Transparência de Desempenho: Permite a re-configuração do sistema para melhorar seu desempenho sem que os usuários percebam.

Transparência de Escalabilidade: Permite que o sistema seja expandido sem que haja a necessidade de alterar sua estrutura ou os algoritmos das aplicações.

As técnicas de transparência de maior importância são a de acesso e localização, pois a presença ou ausência delas afeta fortemente a utilização dos recursos (COULOURIS, 2005). Como exemplo de uma transparência de acesso, pode-se considerar uma interface onde o usuário possa acessar seus arquivos e diretórios locais e remotos, enquanto que um exemplo da falta da transparência de acesso seria um sistema distribuído que não permitisse o acesso aos arquivos em um computador remoto a menos que o cliente faça uso, por exemplo, de um aplicativo de FTP (*File Transfer Protocol*) (COULOURIS, 2005).

4.4 FERRAMENTAS DE SISTEMAS DE ARQUIVOS DISTRIBUÍDOS

Ao longo deste capítulo são apresentadas algumas das principais ferramentas de SAD do mercado.

4.4.1 Coda file system

O *Coda* começou a ser desenvolvido em 1987 pela Universidade de Carnegie-Mellon e é descendente da versão 2 do AFS, do qual herdou muitas de suas características arquitetônicas. Seu principal objetivo é fornecer operações desconectadas ao sistema de arquivos para computadores portáteis, que costumam ficar grande parte do tempo fora da rede. Isso provê uma máxima disponibilidade dos arquivos aos usuários, pois os arquivos residirão na máquina local, evitando assim que transtornos como, por exemplo, a queda da rede ou do serviço de armazenamento distribuído, venha a causar contratempo para o usuário. Além da operação desconectada, o *Coda* utiliza replicação de arquivos como meio de aumentar a disponibilidade (KON, 1994; TANENBAUM, 2002).

O suporte a operações desconectadas surgiu com a proposta de criar um sistema de arquivos resistente a falhas de rede entre servidores e clientes. O esforço empregado no *Coda* para lidar com este problema mostrou-se conveniente com o advento de clientes móveis.

Quando um usuário atualiza um arquivo, as alterações precisam ser propagadas aos servidores que o armazenam. Caso o cliente esteja conectado ao servidor, esta atualização é feita de forma síncrona, ou seja, a alteração é propagada no momento de sua gravação no cliente.

Caso haja problemas nesta comunicação, ao invés de reportar o erro ao usuário, as informações são gravadas localmente em um registro de alterações pendentes e assim que a comunicação com os servidores for restabelecida, estas alterações serão propagadas automaticamente (KISTLER, 1991; BRAAM, 1998).

Os clientes do Coda possuem o sistema de arquivos remoto montado abaixo de */coda*, similar ao seu antecessor. Qualquer arquivo compartilhado por algum servidor Coda, estará disponível neste ponto de montagem para todos os clientes. Diferente do NFS os usuários conectam-se ao serviço Coda e não individualmente nos servidores, que são adicionados como membros do serviço de forma transparente (BRAAM, 1998).

Ao contrário do AFS, no *Coda* os volumes de dados podem possuir várias cópias que podem ser tanto lidas quanto alteradas pelos clientes. Cada volume possui um grupo de servidores que os replicam, o *Volume Storage Group* (VSG). O subconjunto de servidores do VSG que estão acessíveis por um usuário em um determinado instante é o *Accessible Volume Storage Group* (AVSG) do cliente. Quando o usuário envia para o servidor um arquivo modificado, as alterações são propagadas, em paralelo, para todos os servidores do AVSG e posteriormente, para os demais servidores do VSG (BRAAM, 1998; KON, 1994; TANENBAUM, 2002).

Quando um servidor reintegra a rede, nenhuma providência é tomada para a atualização dos arquivos. Os arquivos somente serão atualizados neste servidor caso seja realizada uma requisição de acesso a uma versão mais recente. Se o cliente descobre que o seu servidor preferencial está com uma versão antiga do arquivo, o cliente solicita a versão mais recente de quem a possui e emite uma mensagem ao AVGS informando da existência de uma versão desatualizada (KON, 1994; TANENBAUM, 2002).

4.4.2 Network file system - NFS

Desenvolvido pela Sun Microsystem, este sistema de arquivos foi projetado e implementado para uso em estações de trabalho que rodam *Unix*. Atualmente é utilizado por outros fabricantes, tanto em estações *Unix* quanto aquelas que rodam outros sistemas operacionais e é considerado uns dos principais sistemas de arquivos distribuídos.

4.4.2.1 Arquitetura - Network File System

Na *interface* cliente, a transparência do NFS ocorre através de um sistema de arquivos virtual (*Virtual File System* - VFS), que consiste em atribuir as requisições ao módulo de sistemas de arquivos apropriado, agindo como um *middleware* – camada de *software* que faz intermédio na comunicação entre outras aplicações – para as aplicações do usuário. Isso possibilita trabalhar com o sistema de arquivo local, com o módulo do NFS ou outro sistema de arquivos (COULOURIS; DOLLIMORE; KINDBERG, 2005).

O serviço de arquivo remoto do NFS emprega o uso de um modelo de acesso remoto, que realiza as operações sobre uma interface que contém as várias operações disponíveis para a aplicação local. Contudo, o servidor é o responsável em aplicar tais operações (TANENBAUM; STEEN, 2007).

O NFS foi projetado para funcionar sem manter o estado do sistema, ou seja, o servidor não mantém quais são os arquivos abertos pelos clientes. Isso simplifica todo o processo, pois, caso ocorra uma queda do servidor, não será necessário recuperar as informações. No lado do cliente não será necessário conhecer a atual situação do servidor, apenas que ele continue enviando as solicitações dos dados até que o servidor responda (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Na sua quarta versão, além do suporte para trabalhar sem informações de estado, existe a alternativa de trabalhar com operações que mantêm estados em conjunto com o cliente, permitindo que ambos saibam sobre as operações realizadas

em determinados arquivos. Isso auxilia no modo como a *cache* irá trabalhar, além de possibilitar o uso de travas (*locks*) nativamente para o bloqueio de blocos de arquivos e a realização de operações atômicas. Outra característica foi a substituição do protocolo de transporte UDP (*User Datagram Protocol*) pelo TCP (*Transfer Control Protocol*) na comunicação entre servidores e clientes, possibilitando confirmar o recebimento das mensagens (SHEPLER et AL, 2000, 2003).

A organização do espaço de nomes procede da utilização da hierarquia de diretórios UNIX através do serviço de montagem. Em cada servidor há uma lista de acesso contendo os nomes dos diretórios e quem poderá montá-los. No lado do cliente eles são montados utilizando o comando *mount* (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Para facilitar o processo de montagem dos diretórios, foi acrescentado ao cliente NFS o recurso *automounter*, que armazena os caminhos dos servidores. Quando o usuário busca resolver um determinado caminho de acesso a um arquivo no servidor, é encaminhado ao *automounter* que procura em sua lista o sistema de arquivos solicitado e envia um pedido de montagem do sistema de arquivos remoto (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Uma visão geral da arquitetura do NFS está ilustrada na Figura 1.

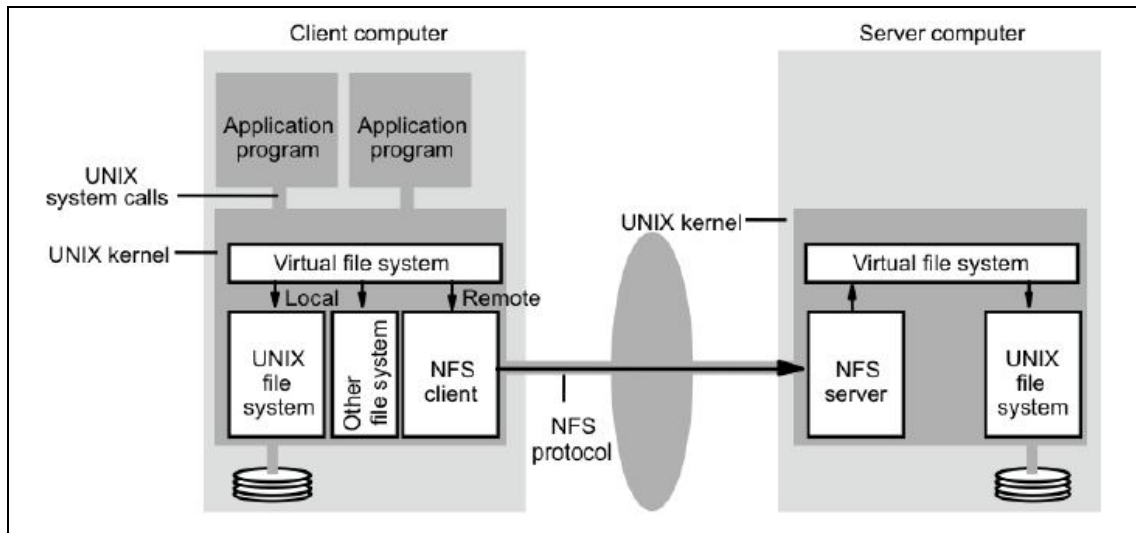


Figura 1 - Arquitetura AFS.
 Fonte: Tanenbaum e Steen (2007).

4.4.2.2 Cache – Network File System

No lado do servidor, *cache* armazena blocos de arquivos, diretórios e atributos de arquivos que foram lidos do disco. Uma forma de agregar maior agilidade à *cache* é através do método *read-ahead*, que antecipa quais serão os próximos blocos a serem lidos e já procura carregá-los na memória.

Há também a utilização da técnica *delayed-write*, o qual consiste em adiar a escrita dos dados nos servidores. A gravação ocorre somente após um período de tempo sem alterações pelo cliente, evitando constantes acessos de escrita a uma curta fração de segundos ao mesmo arquivo. E por fim, a operação *sync* que copia para o disco os blocos alterados a cada 30 segundos (KON, 1994), (COULOURIS; DOLLIMORE; KINDBERG, 2005).

No cliente, a *cache* é usada para armazenar suas operações e com isso diminuir os acessos aos servidores. Entretanto, isso pode causar problemas de consistência nas *caches* dos demais arquivos distribuídos por outros clientes. Para isso, antes dele utilizar os dados que estão em sua *cache*, ele busca no servidor,

através de um processo baseado em verificação de *timestamps*, que atualiza os arquivos da sua *cache* para a versão mais atual.

4.4.2.3 Segurança -Nnetwork File System

Os esforços para proporcionar segurança ocorreram através de um canal de comunicação seguro entre cliente e servidor. Para isso, ele fornece suporte ao RPCSEC_GSS (EISLER et al,1997) que expressa uma grande quantidade de mecanismos de segurança no auxílio à proteção dos canais de comunicação e suporte a integridade e confidencialidade das mensagens. O RPCSEC_GSS atua como uma camada sobre as interfaces de segurança que, entre elas, trás o suporte ao Kerberos v5 (KOHL, 1993) e ao método de chaves públicas conhecido como Lipkey (EISLER, 2000).

O desenvolvimento do NFS seguiu o critério de não utilizar mecanismos de segurança próprio, em vez disso, optou em apenas padronizar o modo para a manipulação da segurança. Com isso novos mecanismos de segurança podem ser incorporados ao NFS. Enquanto isso, o controle de acesso ocorre através da lista de controle de acesso (*Access Control List - ACL*) que especifica as permissões para o usuário ou grupo (TANENBAUM, STEEN, 2007).

4.4.3 Andrew File System - AFS

O projeto *Andrew File System* (AFS) começou na Universidade Carnegie-Mellon em 1983, com o apoio da IBM, e visava o desenvolvimento de um sistema que fosse ideal para o ambiente acadêmico de ensino e pesquisa, oferecendo a cada aluno e professor uma estação de trabalho com um sistema compatível com o Unix BSD, onde os usuários entrariam em qualquer máquina da rede e sua visão do sistema deveria ser a mesma. Desejava-se estender esta transparência de localização a uma rede de 5 a 10 mil estações de trabalho, logo, era essencial tomar o máximo de cuidado com a escalabilidade dos métodos empregados, pois um

sistema que se comporta perfeitamente em uma rede experimental de 50 máquinas, pode mostrar-se completamente inadequado para uma rede 100 vezes maior (KON, 1994).

Em termos de arquitetura, o AFS possui alguns pontos em comum com o NFS, como o uso da interface VFS (*Virtual File System*) para os clientes possuírem acesso aos arquivos e o uso do RPC como mecanismo de comunicação, o qual utiliza o formato XDR para a representação externa de dados. Outros componentes que fazem parte da arquitetura do AFS são o *Vice* e o *Venus*. *Vice* é o nome dado ao processo servidor que atende as solicitações de serviços de arquivos que chegam. Desde a versão 2 do AFS, o *Vice* é *multi-threaded*, e desta forma, consegue atender a várias solicitações simultâneas eficientemente. O *Venus* é o nome dado ao cliente AFS, o qual é responsável pela interface entre o cliente e o *Vice*. A Figura 2 apresenta a distribuição dos processos *Vice* e *Venus* (KON, 1994; COULOURIS, 2005).

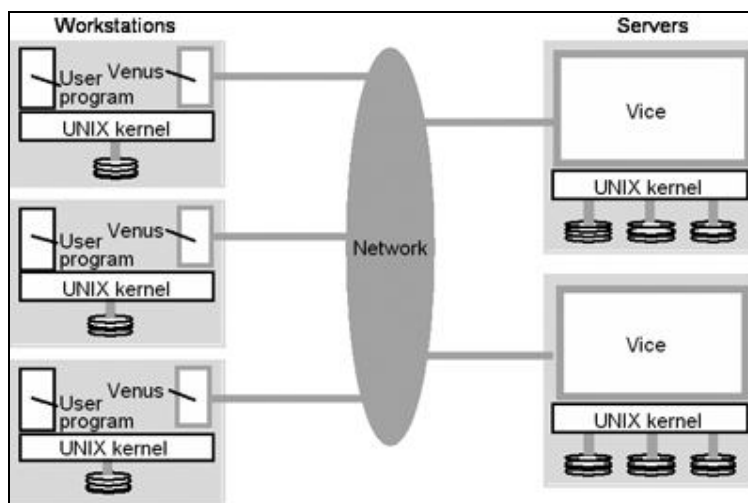


Figura 2 - Distribuição dos processos no Andrew File System.
Fonte: Coulouris (2005).

O espaço de nomes do AFS é dividido em duas partes. A primeira é armazenada em discos locais e guardam basicamente informações temporárias (diretório */tmp*), o *cache* do cliente (diretório */cache*) e os arquivos necessários para a inicialização da máquina. A segunda parte (diretório */afs*) é igual para todas as máquinas da rede e contém os arquivos compartilhados mantidos por servidores dedicados (KON, 1994). A Figura 3 exemplifica o espaço de nomes de um cliente.

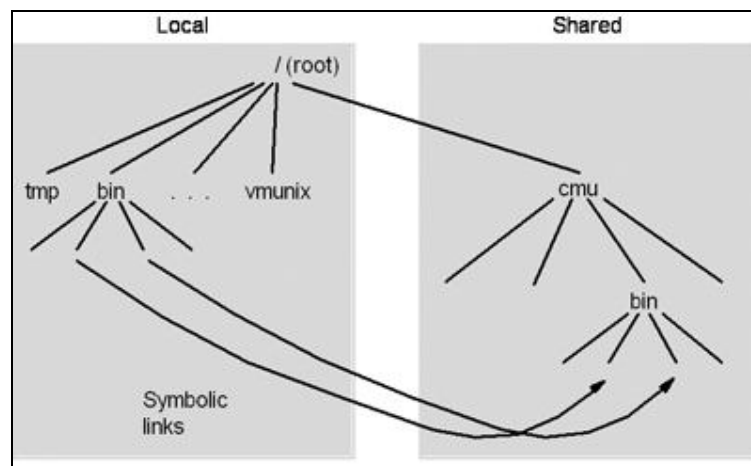


Figura 3 - Espaço de nomes vistos de um cliente AFS.
Fonte: Coulouris (2005).

O AFS disponibiliza recursos para a replicação de volumes e para a realização de cópias de segurança, onde apenas uma das réplicas é considerada alterável, enquanto as demais funcionam apenas para leitura. Um cliente AFS sempre acessa a cópia do volume que está mais próxima. Caso um servidor não responda ao chamado de um cliente, este passa automaticamente a acessar os mesmos dados residentes em outra cópia ativa (KON, 1994).

4.4.4 Distributed File System - DFS

O *Distributed File System* faz com que múltiplos usuários compartilhem arquivos e recursos tendo como objetivo gerenciar recursos de forma eficiente mantendo-os disponíveis e seguros para os usuários, sua principal característica é a capacidade de gerenciar vários dispositivos de armazenamento, que se encontram dispersos em diferentes localidades.

Com a ferramenta, arquivos distribuídos a diversos servidores de um domínio podem ser acessados de forma transparente podendo ser conectados a um ou mais *namespaces* DFS, simulando que todo o material reside em um único servidor. Isso elimina a necessidade de os usuários acessarem vários locais da rede para localização de que precisam. A *namespace* DFS é uma exibição virtual das pastas compartilhadas em uma organização (MICROSOFT, 2013).

O DFS permite que administradores do sistema facilitem para os usuários o acesso e o gerenciamento de arquivos distribuídos fisicamente em uma rede. Com o DFS, você pode fazer com que arquivos distribuídos em vários servidores pareçam residir em um local na rede. Os usuários não precisam mais conhecer e especificar o local físico real dos arquivos para acessá-los. Quando um usuário exibe o *namespace*, as pastas parecem residir em um único disco rígido de alta capacidade. Os usuários podem navegar no *namespace* sem precisar saber os nomes dos servidores ou pastas compartilhadas que hospedam os dados. DFS também oferece muitos outros benefícios, incluindo tolerância a falhas e recursos de compartilhamento de carga, recursos que o tornam ideal para organizações que necessitam de um alto nível de disponibilidade a seus arquivos de dados (MICROSOFT, 2013).

4.4.5 Google File System

O *Google File System* é um sistema que foi projetado e implementado para uso interno no *Google*, para ambientes que lidam com milhares de arquivos muito extensos – que podem variar de algumas centenas de *megabytes* a vários *gigabytes* – e onde operações de leitura ou de adição (*append*) de arquivos são predominantes (GHEMWAT, 2003).

Foram considerados certos aspectos mais proeminentes durante a análise da operação dos servidores pertencentes à *Google Inc.* São esses:

Falhas no sistema: Considerando que o *hardware* utilizado pela empresa é composto em sua maioria de sistemas de baixo custo (i.e.: computadores comuns, como PCs), falhas em componentes individuais do sistema acontecem em uma taxa relativamente alta e um tanto quanto regular, porém são considerados naturais, e de forma alguma inesperados. Com isso em mente, o sistema deve ser largamente capaz de identificar qualquer falha (auto-monitoramento) e se recuperar de forma automática, a qualquer instante.

Tamanho dos arquivos: A maioria dos arquivos no sistema tem um tamanho que varia de algumas centenas de *megabytes* a alguns *gigabytes*, sendo mais comuns arquivos com vários *gigabytes*. Arquivos pequenos são suportados, porém, o GFS não é otimizado para esses casos de predominância de arquivos de tamanho reduzido.

Leitura de arquivos: A leitura no *Google File System* pode ser dividida em dois tipos: o primeiro tipo consiste na leitura de fluxos de dados maiores, geralmente múltiplos *megabytes*, sendo que se o mesmo cliente solicita mais de uma leitura sucessiva, os dados usualmente estão armazenados de forma contínua no disco. O segundo tipo engloba leituras menores (alguns poucos *kilobytes*) e em regiões aleatórias. Pressupõe-se que aplicações levem esses aspectos em consideração e organizem suas leituras menores para que sejam efetuadas em seqüência e em lote.

Escrita de arquivos: As escritas geralmente consistem em grandes fluxos de dados (aproximadamente do mesmo tamanho das leituras do primeiro tipo) que são adicionadas a arquivos (*appends*). Após isso, os arquivos raramente são modificados. Também pode haver escritas de pequenos trechos arbitrários em um determinado arquivo.

Concorrência: O sistema deve ser capaz de fornecer uma semântica bem definida quanto às operações de escrita realizadas de forma concorrente por múltiplos clientes no mesmo arquivo. Isso implica na essencialidade de se fornecerem operações de escrita atômicas, com o mínimo possível de *overhead*.

Utilização da rede: Largura de banda e tempo de disponibilidade da conexão são preferidos a baixas latências. A maioria das aplicações que executam sobre o sistema costumam priorizar o processamento de grandes blocos de dados a altas taxas ao invés do tempo de resposta reduzido.

Quando um cliente necessita de um arquivo, ele entra em contato com o servidor principal que o orienta informando em qual computador está localizado o arquivo solicitado. A partir daí, todas as alterações realizadas pelo cliente no arquivo solicitado são encaminhadas diretamente ao computador indicado pelo servidor principal. Além deste apontamento, o servidor principal é responsável por gerenciar a replicação dos arquivos e coordenar os processos de inclusão e remoção de computadores na rede (GHEMAWAT, 2003).

5 METODOLOGIA

Foi desenvolvida uma pesquisa exploratória, pois de acordo com Gil (2008), o objetivo desta é familiarizar-se com um assunto ainda pouco conhecido, pouco explorado. Ao final de uma pesquisa exploratória, se conhecerá mais sobre aquele assunto, estando apto a construir hipóteses. Como qualquer exploração, a pesquisa exploratória depende da intuição do explorador (neste caso, da intuição do pesquisador).

Após toda pesquisa exploratória, o trabalho implementado utilizou um equipamento notebook do fabricante Acer com processador *Intel Core i3*, 6gb de memória RAM DDR3, disco rígido de 500 Gb, com *Windows Seven Professional* 64 bits como Sistema Operacional.

Para demonstrar cada servidor virtualizado e a implementação do trabalho, foi criado um usuário e dois servidores Windows, cuja máquina virtual que foi utilizada pelo usuário foi de configuração Windows XP, gerenciados pelo software Oracle VM VirtualBox 4.2.8-83876. Cada servidor virtualizado foi instalado o Microsoft *Windows Server 2008 R2 Enterprise*.

No VM VirtualBox foi escolhida a rede interna entre as máquinas virtuais, não havendo interação com a máquina física. Todas as máquinas tiveram seu IP fixo.

Foi configurado o DFS em ambos servidores virtualizados, onde podemos mostrar a replicação dos arquivos criados pelo usuário do Windows XP fazendo com que os arquivos pudessem ser acessados e salvos de forma segura, transparente, consistente e tolerante a falhas.

O objetivo desse trabalho foi simular um cenário onde o usuário tem acesso ao domínio, podendo criar, salvar, excluir, modificar os arquivos mesmo com o desligamento de um dos servidores, mostrando a eficiência do sistema de arquivos distribuídos e fazendo com que o usuário não perceba o que tenha acontecido.

6 RESULTADOS

6.1 IMPLEMENTAÇÃO DO DFS

Levando em consideração os tópicos abordados nas seções anteriores, o SAD foi implementado junto com os conceitos e técnicas estudados.

Para a criação do ambiente foi preciso à instalação e configuração dos sistemas e rede abaixo conforme Figura 4:

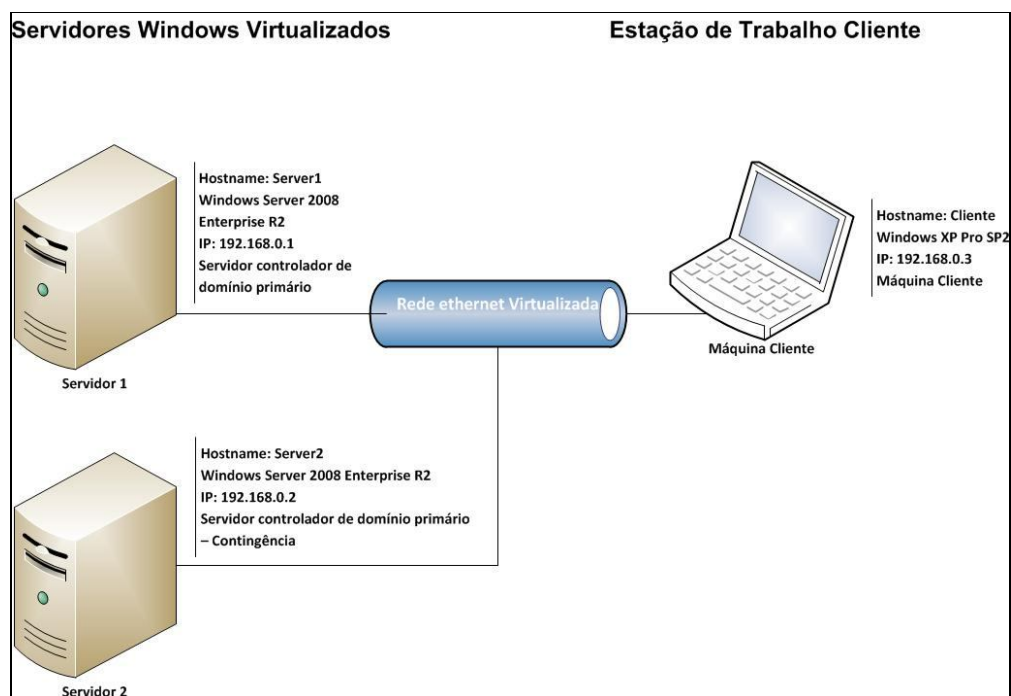


Figura 4 - Ambiente virtualizado.
Fonte: Elaborada pela autora (2013).

Para que os sistemas fossem executados sem erros, foi necessária a verificação da configuração mínima. Os servidores virtuais tiveram 2gb de memória RAM, 25gb de disco rígido e a máquina cliente recebeu 256mb com 10gb de disco.

Vemos os servidores e a máquina cliente em execução na Figura 5 e Figura 6.

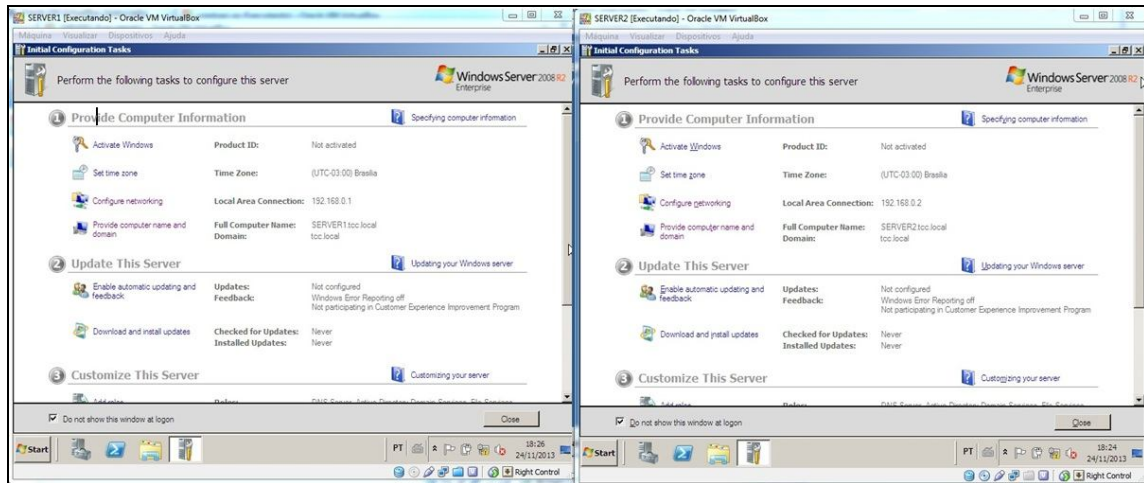


Figura 5 - Máquinas virtuais dos servidores em execução.
Fonte: Elaborada pela autora (2013).

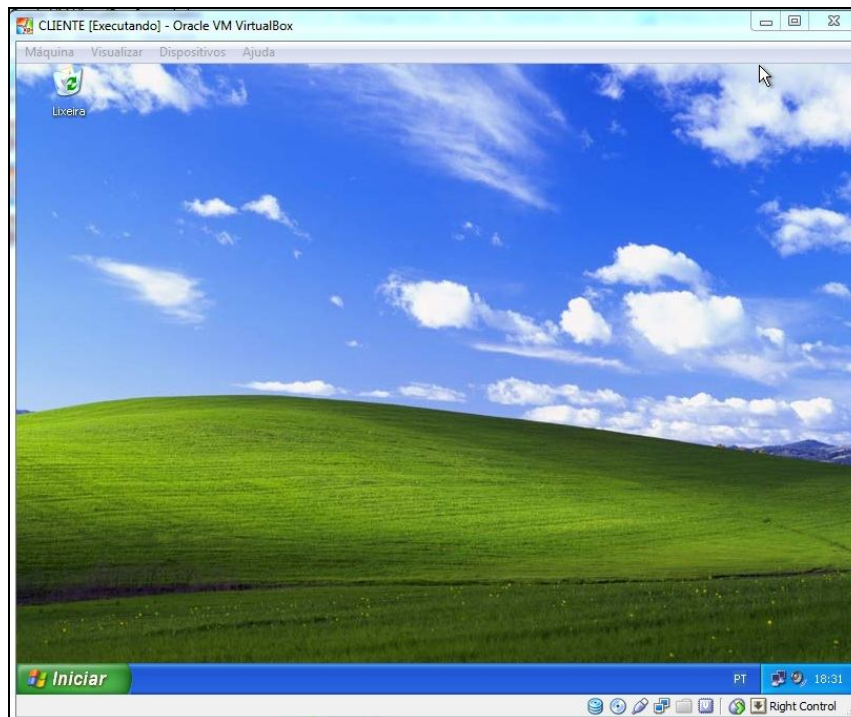


Figura 6 - Máquina virtual do cliente.
Fonte: Elaborada pela autora (2013).

Após a instalação do *Windows Server*, o DFS teve que ser habilitado em ambos servidores. Com o “*Server Manager*” aberto tivemos que adicionar e habilitar a função “*File Services*” e marcar os módulos do DFS (*Namespace e Replication*). A Figura 7 mostra a configuração do servidor de arquivos.

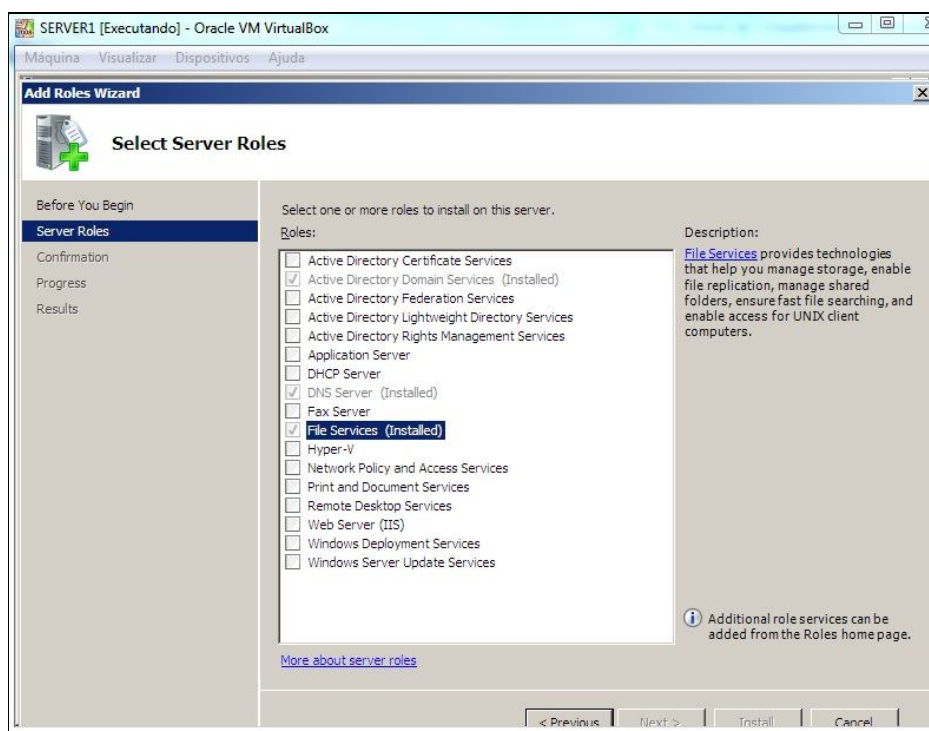


Figura 7 - Configuração do servidor de arquivos.

Fonte: Elaborada pela autora (2013).

Para que a transparência de localização fosse possível, foi habilitada também a função do *Active Directory Domain Services (ADDS)*, para criação do domínio da rede, transformando o SERVER1 em um controlador de domínio primário e o SERVER2 em controlador de domínio backup, e após a criação do domínio: “tcc.local”, para que o usuário possa acessar as pastas sem que consiga identificar a localização física do arquivo. Ou seja, o usuário irá acessar o domínio onde pode

criar os arquivos, modificar, excluir e poderá acessar outras pastas disponíveis a ele. As configurações foram feitas em ambos os servidores, caso o servidor primário torne-se indisponível. A Figura 8 mostra o acesso ao diretório do domínio criado.

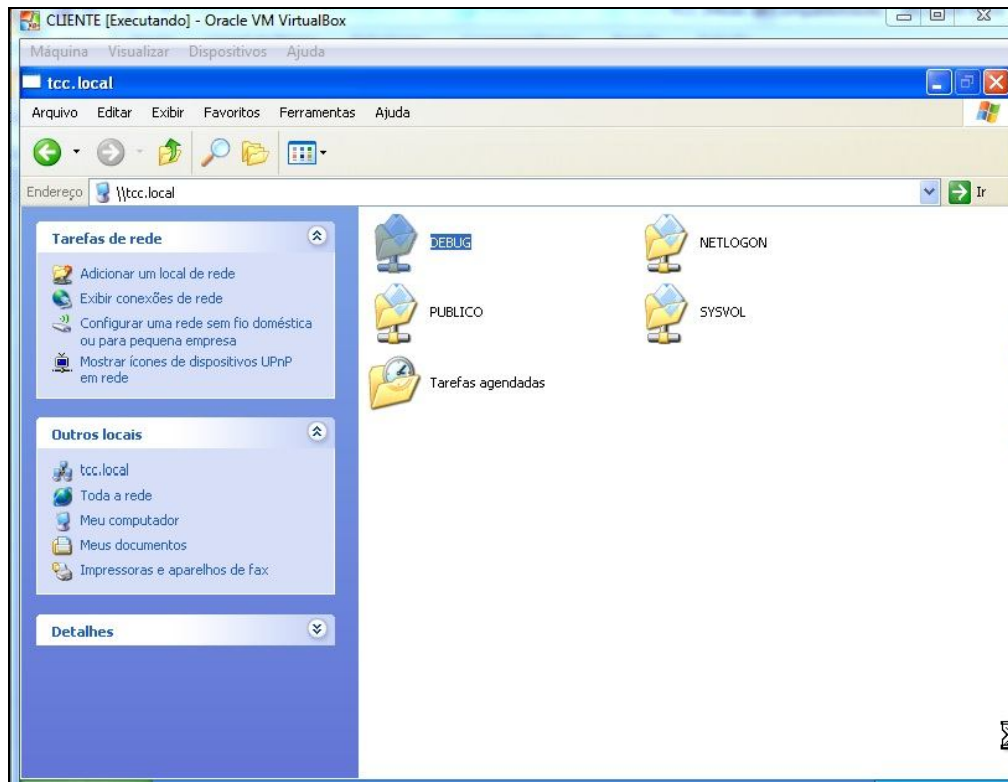


Figura 8 - Acesso ao diretório do domínio.
Fonte: Elaborada pela autora (2013).

Foi necessário criar um novo *namespace* do DFS dentro da guia “*DFS Management*”. O *namespace* permite que agrupe pastas compartilhadas localizadas em diferentes servidores em um ou mais *namespaces* estruturados logicamente. Cada *namespace* aparece aos usuários como uma única pasta compartilhada com uma série de subpastas. Entretanto, a estrutura subjacente do *namespace* pode consistir de várias pastas compartilhadas que se localizam em diferentes servidores e em vários lugares. A Figura 9 mostra a tela de adição de pastas ao *Namespace*.

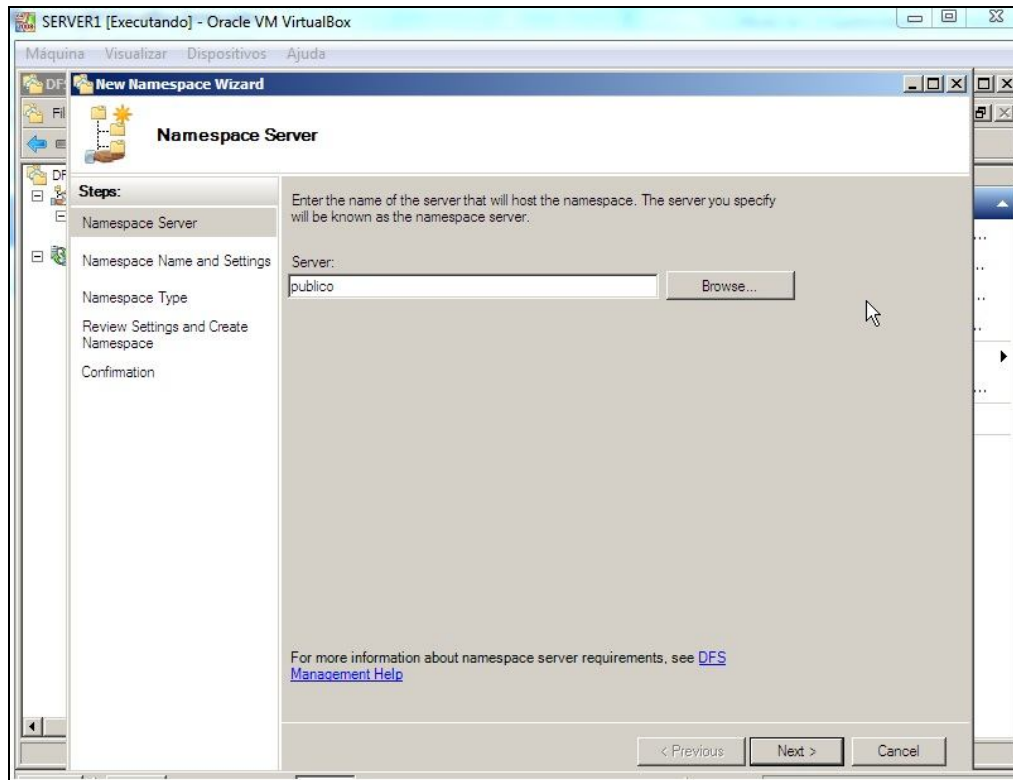


Figura 9 - Adição de um novo *Namespace*.
Fonte: Elaborada pela autora (2013).

Foi configurado um novo Grupo de Replicação do DFS para os diretórios do *Namespace* criados. A Replicação DFS usa muitos processos sofisticados para manter os dados sincronizados em vários servidores. Esta configuração também é realizada no “*DFS Management*” através de um assistente, onde podemos visualizar na Figura 10 a configuração de Replicação do DFS. Podem-se ter diversas configurações disponíveis tais como os servidores membros do grupo (no mínimo dois servidores), a topologia com que será configurada a replicação, largura de banda utilizada, dias e horários específicos para a replicação, membro primário (que terá seus dados replicados para outros servidores) e por fim o diretório local que receberá a replicação física dos arquivos no membro secundário de replicação. Na Figura 10 podemos visualizar a configuração de Replicação do DFS.

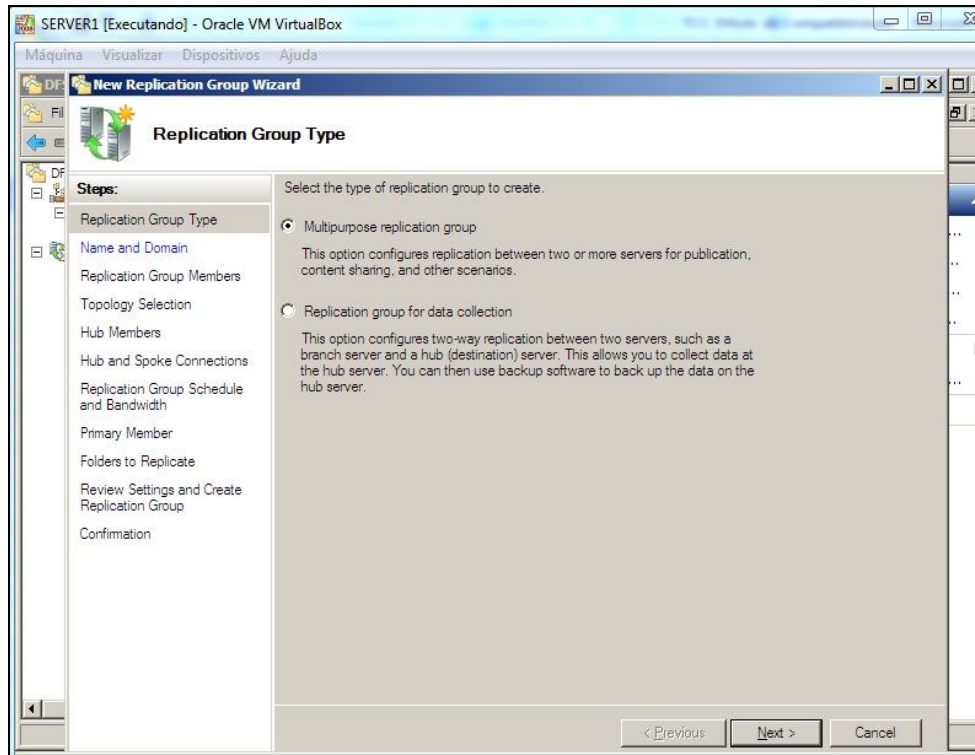


Figura 10 - Configuração da replicação do DFS.
Fonte: Elaborada pela autora (2013).

Com todas as características necessárias para o funcionamento do DFS e após todas as configurações serem um sucesso, nosso ambiente está criado para a realização dos testes de transparência.

6.2 TRANSPARÊNCIA DE ACESSO

Para que a transparência de acesso aconteça, há a necessidade de que o usuário saiba o domínio que irá acessar. Em nosso caso, o usuário acessa via mapeamento de pasta, o usuário não consegue identificar a localização física de

onde seus arquivos estão sendo salvos. O diretório mapeado foi o diretório virtual criado anteriormente pelo *Namespace*, que concede acesso às pastas do DFS, o usuário terá a impressão de que seus arquivos estão em seu computador. Nas Figuras 11 e 12 verifica-se como ficou o mapeamento do diretório “publico” no cliente, que concede acesso ao DFS.

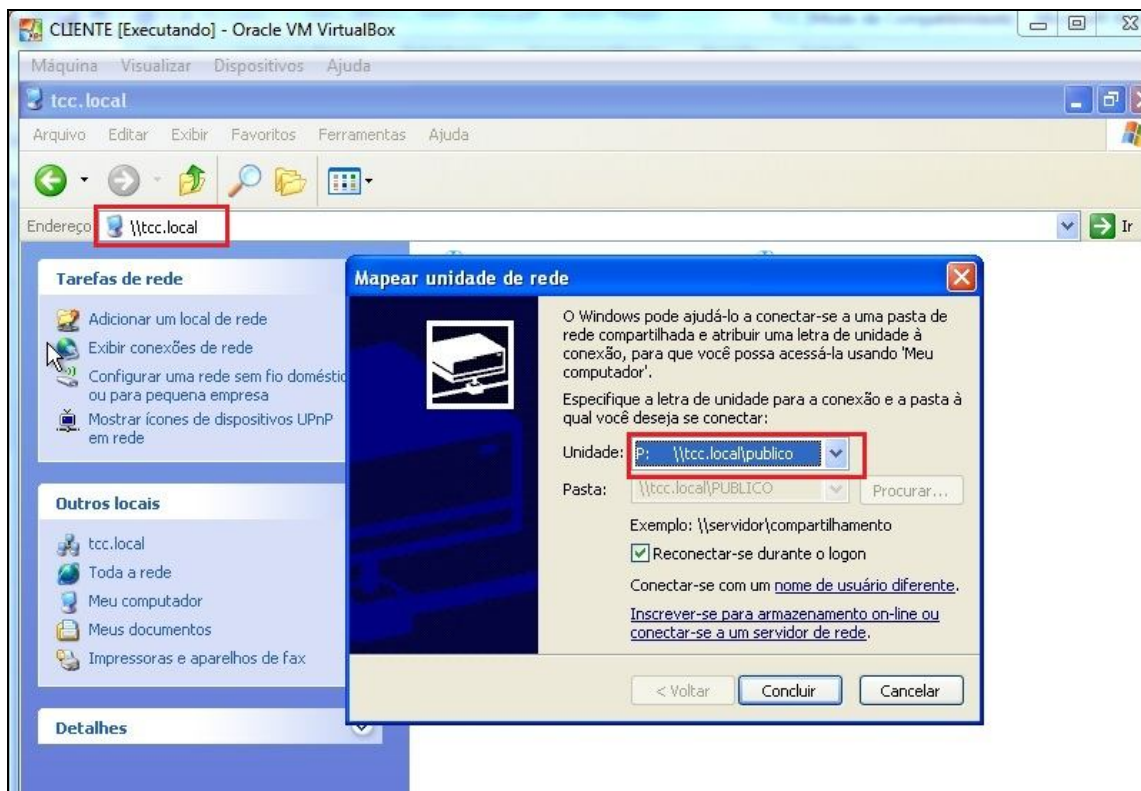


Figura 11 - Mapeamento da unidade dentro do DFS.
Fonte: Elaborada pela autora (2013).

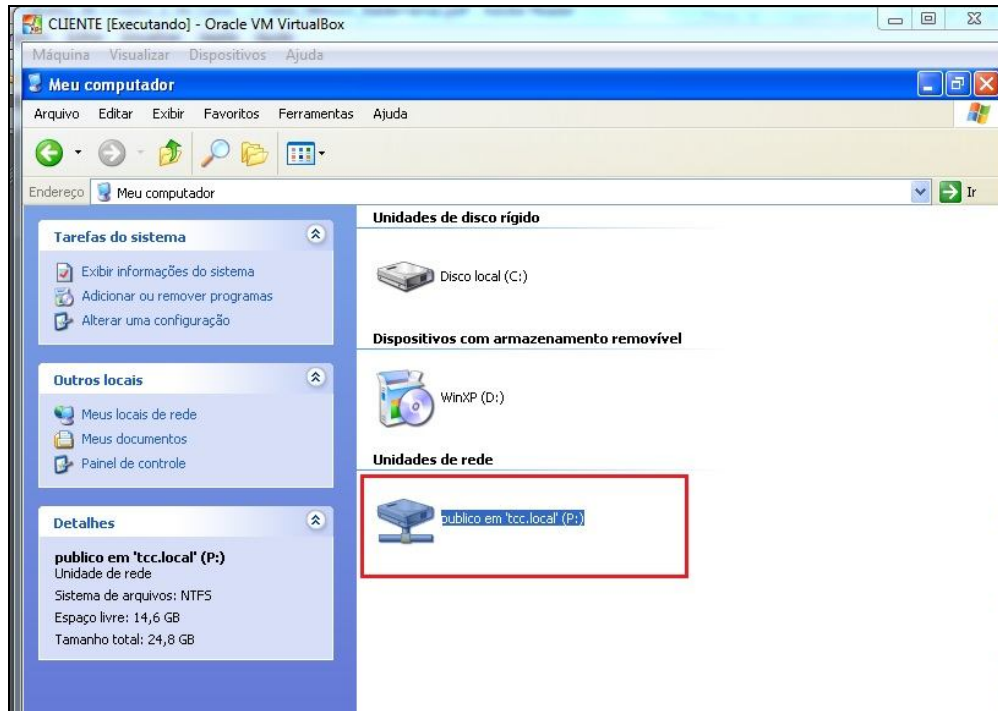


Figura 12 - Mapeamento criado.
Fonte: Elaborada pela autora (2013).

Com a criação do mapeamento foi possível concluir que nosso ambiente preparado teve o objetivo alcançado, onde o usuário conseguiu acessar, criar, modificar, excluir perfeitamente seus arquivos do DFS, como qualquer outra pasta local de seu computador.

6.3 TRANSPARÊNCIA DE REPLICAÇÃO E LOCALIZAÇÃO

Foi testada a transparência de replicação e localização, a partir de que um documento era criado, o usuário não saberia sua localização e não saberia que existiam cópias em outro servidor, servindo assim de *backup*. Em ambos os

servidores foram configurados e ativados a replicação de DFS. Nas Figuras 13 e 14 podemos ver a replicação do arquivo criado pela máquina cliente nos servidores.

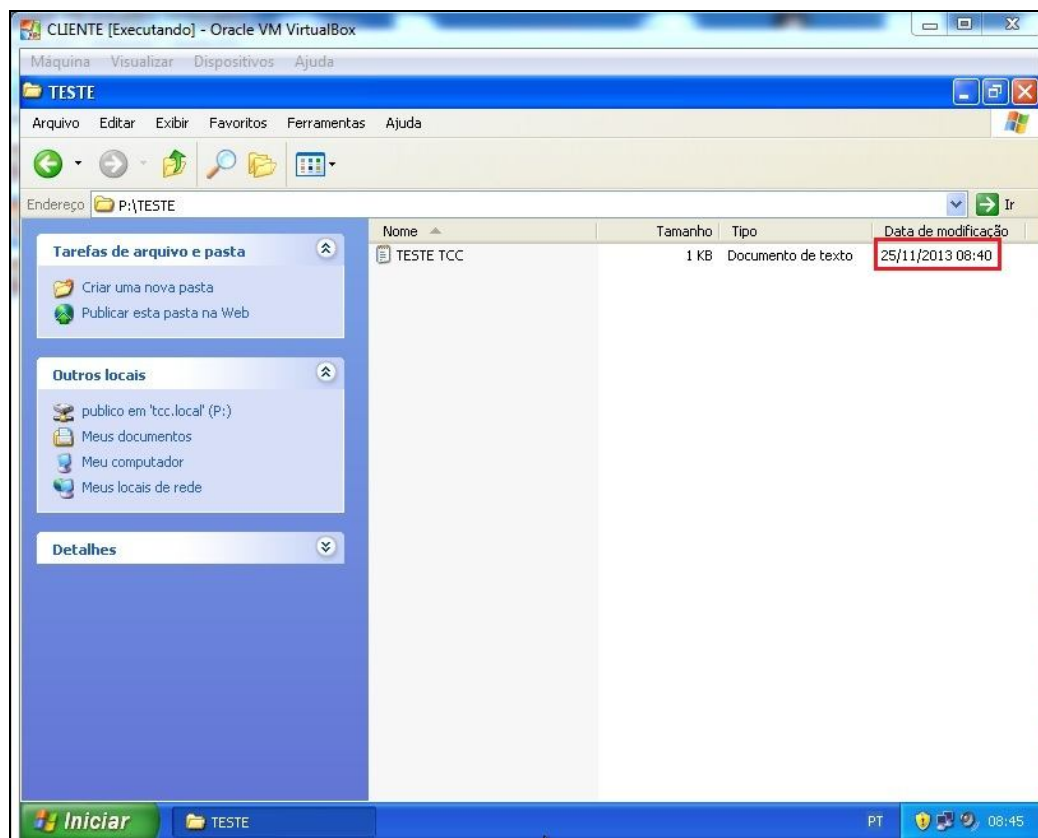


Figura 13 - Criação do arquivo na máquina cliente.
Fonte: Elaborada pela autora (2013).

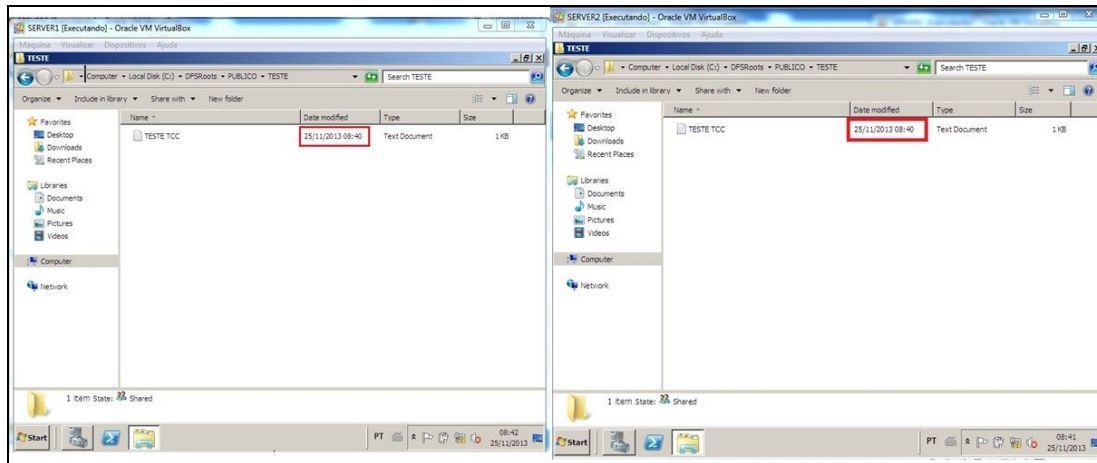


Figura 14 - Replicação do arquivo nos servidores.
Fonte: Elaborada pela autora (2013).

Foi possível verificar que para a criação do arquivo nos servidores é necessário aguardar alguns segundos, após isso a replicação estava concluída, ao simular a indisponibilidade do servidor principal, SERVER1, a máquina cliente fica lenta conforme o servidor *backup*, SERVER2, assumia como contingência, porém após alguns instantes não havia mais oscilação e o usuário ainda conseguia acessar seus arquivos, modificar, excluir, sem qualquer problema.

Mesmo após a indisponibilidade do servidor e pelo fato de que o usuário conseguia acessar os arquivos disponíveis no DFS como se fosse um arquivo em seu computador e desconheça a existência de réplicas de seus arquivos em outros servidores, podemos dizer que este teste foi considerado bem sucedido.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento na área de tecnologia da informação continua intenso e aumentando significativamente, e uma prova disso é o crescente aumento da rede de computadores nas empresas, a necessidade de sistemas de arquivos distribuídos vem crescendo, o que contribui significativamente com a organização e controle dos arquivos das empresas. Atualmente é indispensável que as empresas possuam sistemas que garantam a segurança e disponibilidade das informações sempre que preciso.

O trabalho demonstrou a funcionalidade do DFS no ambiente *Windows Server* 2008, exemplificando uma situação de indisponibilidade do sistema, onde foi desligado o servidor controlador de domínio primário para que assim o servidor controlador de domínio *backup* pudesse assumir sua função, e mostrando que através do sistema de arquivos distribuídos o usuário pode acessar seus arquivos mesmo com a falha, o que torna um ambiente em que trabalha mais seguro, estável e confiável para uma empresa, pois a perda de informações devido falhas no sistema pode causar grandes problemas às organizações, o que não pode ser permitido em um mundo cada vez mais competitivo, onde a tecnologia e sucesso caminham juntos.

Os testes realizados foram importantes, pois através deles foi possível verificar o comportamento prático do DFS, desde sua instalação e configuração, onde o sistema de arquivo distribuído se mostrou muito estável e de fácil manuseio. A pesquisa tanto bibliográfica quanto pratica foi de extrema importância para o desenvolvimento deste trabalho, pois possibilitou atingir os objetivos e veracidade de sistemas de arquivos distribuídos nas organizações.

REFERÊNCIAS

BRAAM, Peter J. **The Coda Distributed File System**. Linux Journal, Seattle, v. 1998, n. 50es, 6 p., Jun. 1998.

CHOW, Randy; JOHNSON, Theodore. **Distributed operating systems and algorithms**. Reading, Massachusetts: Addison Wesley Longman, 1998. 569 p.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Distributed systems: concepts and design**. 4. ed. Harlow: Addison Wesley Longman, 2005. 927 p.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas Distribuídos: Conceitos e Projeto**. 4. ed. Bookman, 2007. 784 p.

EISLER, M. *et al.* **RFC 2203: RPCSEC_GSS Protocol Specification**. Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. sep. 1997. Disponível em: <<http://www.ietf.org/rfc/rfc2203>>. Acessado em: out. 2011.

EISLER, M. **RFC 2847: LIPKEY – A Low infrastructure Public Key Mechanism Using SPKM**. Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. jun. 2000. Disponível em: <<http://www.ietf.org/rfc/rfc2847>>. Acessado em: out. 2011.

GALLI, Doreen L. **Distributed Operating Systems: Concepts and Practice**. New Jersey: Prentice Hall, 2000. 463 p.

GALLO, Michael A.; HANCOCK, William M. **Comunicação Entre Computadores e Tecnologias de Rede**. Brasil: Thomson, 2003. 673 p.

GHEMAWAT, Sanjay; GOBIOFF, Howard; LEUNG, Shun-Tak. **The Google File System**. In: ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 19., 2003, New York. Anais... New York: ACM Press, 2003. p. 29-43.

GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2008.

KISTLER, James J.; SATYANARAYANAN M. **Disconnected operation in the Coda file system**. In: ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 13., 1991, New York. Anais... New York: ACM Press, 1991, p. 213-225.

KOHL, J.; NEUMAN, C. **RFC 1510: The Kerberos Network Authentication Service (V5)**. Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. set. 1993. Disponível em: <<http://www.ietf.org/rfc/rfc1510>>. Acessado em: out. 2011.

KON, Fabio. **Sistemas de Arquivos Distribuídos**. São Paulo: 1994. 154 p. Tese (Mestrado em Matemática Aplicada) - Instituto de Matemática e Estatística, Universidade de São Paulo, 1994.

LAMPORT, L. 1978. **Time, Clocks, and the ordering of events in a distributed system**. Commun., ACM, 21:558-565.

MACHADO, Berenger; MAIA, Luiz Paulo. **Arquitetura de sistemas operacionais**. Rio de Janeiro : Livros Técnicos e Científicos Editora, 1997.

MICROSOFT. **Visão geral sobre o Sistema de Arquivos Distribuídos**. Disponível em: <[http://technet.microsoft.com/pt-br/library/cc738688\(v=ws.10\).aspx](http://technet.microsoft.com/pt-br/library/cc738688(v=ws.10).aspx)>. Acesso em: 16 maio 2013

MICROSOFT. **O que é o DFS?**. Disponível em: < <http://technet.microsoft.com/en-us/library/cc779627.aspx&usg=ALkJrhgqOtxNSI7g6IO3KDfQfCjWuDNA>>. Acesso em: 16 maio 2013.

MICROSOFT. **Como DFS funciona**. Disponível em: < http://technet.microsoft.com/en-us/library/cc782417.aspx&usg=ALkJrhjNG_GCBh6fO4ITd19xL18wJ5wH3Q>. Acesso em: 16 maio 2013.

SHEPLER, S. et al. **RFC 3010**: NFS version 4 Protocol. Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. dec. 2000. Disponível em: <<http://www.ietf.org/rfc/rfc3010> >. Acessado em: out. 2011.

SHEPLER, S. et al. **RFC 3530**: Network File System (NFS) version 4 Protocol. Published by Internet Engineering Task Force (IETF). Internet Society (ISOC) RFC Editor. USA. apr. 2003. Disponível em: <<http://www.ietf.org/rfc/rfc3530>>. Acessado em: out. 2011.

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. **Applied Operating System Concepts**. New York: John Wiley & Sons, 2000. 840 p.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer. **Sistemas Operacionais com Java**. Rio de Janeiro: Campus, 2008. 672 p.

SINHA, Pradeep K. **Distributed Operating Systems**: Concepts and Design. New York: Wiley-IEEE Press, 1996. 764 p.

STALLINGS, William. **Cryptography and Network Security**: Principles and Practice. New Jersey: Prentice Hall, 2003, 681 p.

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Rio de Janeiro : Prentice-Hall do Brasil, 1995. 493 p.

TANENBAUM, Andrew S.; STEEN, Maarten van. **Distributed Systems: Principles and Paradigms**. New Jersey: Prentice Hall, 2002. 803 p.

TANENBAUM, Andrew S. **Computer Networks**. New Jersey: Prentice Hall, 2000. 384 p.

TANENBAUM, Andrew S. **Operating systems: design and implementation**. Englewood Cliffs : Prentice-Hall, 1987.

TANENBAUM, Andrew S.; STEEN, Maarten van. **Distributed Systems: Principles and Paradigms**. New Jersey: Prentice Hall, 2002. 803 p.

TANENBAUM, Andrew S.; STEEN, Mararte Van. **Sistemas Distribuídos: princípios e paradigmas**. 2. ed. Prentice-hall, 2007. 416 p.

IMPLANTAÇÃO DE UM SISTEMA DE ARQUIVOS DISTRIBUÍDOS

Thais M. A. Leite, Henrique P. Martins, Marcio H. C. Cardim, Elvio G. Silva.

Instituto de Informática – Universidade Do Sagrado Coração (USC) Bauru – SP

Centro de Ciências Exatas e Sociais Aplicadas– Universidade do Sagrado Coração,
USC.

tha.leite91@gmail.com, henmartins@gmail.com, mcardim@usc.br,
egsilva@usc.br

Abstract. *This article aims to implement a distributed file system (DFS), where was configured Distributed File System (DFS) on virtual servers so that the user can store and manipulate files without to know the physical location of the same, facilitating the process of storing and sharing files safely. The study presents basics on distributed file systems, citing some of the systems that utilize and exemplifying a scenario in DFS, this sets the work of implementation of a distributed file system. The results of this study demonstrate how efficient can be a DFS within an organization.*

Resumo. *Este artigo tem como objetivo à implementação de um sistema de arquivos distribuídos (SAD), onde foi configurado o Distributed File System (DFS) em servidores virtuais para que o usuário possa armazenar e manipular arquivos sem que saiba a localização física dos mesmos, facilitando o processo de armazenamento e compartilhamento de arquivos com segurança. O estudo apresenta fundamentos sobre sistemas de arquivos distribuídos, citando alguns dos sistemas que mais se utilizam e exemplificando um cenário no DFS, isto define o trabalho de implementação de um sistema de arquivo distribuído. Os resultados deste estudo comprovam o quão eficiente pode ser um SAD dentro de uma organização.*

1. INTRODUÇÃO

No começo da década de 70, foram desenvolvidos os disquetes, e devido ao pequeno tamanho dos arquivos era considerado um grande dispositivo de armazenagem e com o passar dos anos foram surgindo novas formas de compartilhar e armazenar arquivos como CDs, DVDs, cartão de memória, Pen Drive, Blu-Ray, além da existência de armazenamento de arquivos, tais como armazenamento distribuído.

Um sistema distribuído consiste de computadores independentes ligados uns com os outros. A diferença básica entre um sistema distribuído e uma rede de computadores é que em um ambiente distribuído os recursos são disponibilizados para o usuário de forma transparente. Isso significa que, teoricamente, os usuários nem sabem que os computadores estão interconectados. (GALLO. 2003)

A utilização de uma DFS é feita em grandes corporações, cuja rede de computadores é uma WAN (*Wide Area Network*), possibilitando assim o gerenciamento dos arquivos compartilhados e replicando arquivos em outro servidor para que haja disponibilidade. Hoje usuários podem criar, acessar, modificar seus arquivos sem se preocupar onde está localizado, pois acessam os arquivos com transparência, segurança e eficiência.

2. SISTEMA DE ARQUIVOS

O sistema de arquivos controla a forma como os arquivos são organizados, como será o acesso aos dados, como serão identificados pelo usuário, quais os atributos e operações possíveis sobre eles.

É através do sistema de arquivos que os usuários terão uma interface para armazenar e recuperar seus dados, de forma transparente quanto aos detalhes de implementação e organização. É através dele também que os diferentes processos do sistema poderão executar tarefas sobre os arquivos ou compartilhá-los com outros processos (TANENBAUM, 1995; MACHADO, 1997).

O sistema de arquivos pode oferecer serviços para atender as necessidades de armazenamento, controle e recuperação de arquivos e também serviços para a organização desses arquivos em diretórios.

2.1 Sistema de arquivos distribuídos

A construção de sistemas distribuídos vem da motivação em compartilhar recursos computacionais de hardware, software e dados, o que envolve alguns problemas como concorrência, inexistência de um relógio global, falhas independentes, dentre outras mais (COULOURIS, 2005).

Vários conceitos importantes devem ser observados para a construção de um sistema distribuído, como a transparência na distribuição em seus diferentes níveis, fornecer acesso fácil aos recursos do sistema, o tratamento eficiente das falhas no sistema quando estas acontecerem, e permitir que o sistema possa ser expandido (TANENBAUM; STEEN, 2007), (COULOURIS, 2005).

Um Sistema de Arquivos Distribuídos (SAD) tem o objetivo de fornecer os mesmos serviços e recursos de um sistema de arquivos convencional, ao menos na ótica dos clientes que os utiliza, com a diferença que este pode ser acessado de qualquer máquina dentro de uma rede (acesso remoto).

2.2 Características desejáveis em um sistemas de arquivos distribuídos

Muitos dos conceitos encontrados na implementação de sistemas distribuídos devem ser levados em consideração no desenvolvimento de sistemas de arquivos distribuídos. Inicialmente, os primeiros sistemas de arquivos distribuídos ofereciam recursos de transparência de acesso e transparência de localização, emergindo subsequentemente à preocupação no desenvolvimento de recursos como desempenho, escalabilidade, controle de concorrência, tolerância a falhas e segurança (COULOURIS, 2005).

2.2.1 Atualização concorrente

A concorrência segundo Coulouris, Dollimore e Kindberg (2007), se caracteriza quando usuários podem acessar vários arquivos, ou os mesmos arquivos de forma simultânea,

sem sofrer danos, perda de desempenho ou outras restrições. Estes acessos devem ser realizados sem que o usuário necessite saber como é realizado pelos servidores. Dessa forma, salienta-se que a transparência de concorrência e de paralelismo é uma importante característica dos Arquivos Distribuídos.

2.2.2 Replicação

Caso um sistema de arquivos tenha suporte à replicação, a confiança e a eficiência do serviço de arquivos são aumentadas de forma significativa. Eficiência em relação a tempo de resposta, carga do servidor e tráfego de rede, conceitos que podem ser verificados em Coulouris, Dollimore e Kindberg (2007). Adicionalmente, se um servidor sair do ar, o serviço de arquivos ainda pode manter suas obrigações.

Desta maneira a replicação de arquivos provém tolerância a falhas, uma vez que o usuário não consegue perceber que o servidor o qual estava conectado caiu e que outro (ou outros) entrou no seu lugar para fornecer o arquivo que ele estava utilizando. Assim é interessante que o sistema também ofereça transparência de replicação, pois não é necessário o usuário saber como o sistema trata da replicação deste arquivo.

2.2.3 Heterogeneidade

O sistema de arquivos distribuído deve ser implementado de forma que o servidor e o cliente não necessitem da mesma arquitetura de hardware e da mesma solução de software para a correta execução do serviço (COULOURIS, 2005; KON, 1994).

2.2.4 Tolerância a falhas

A tolerância a falhas segundo Tanenbaum e Sten (2007) é que mesmo que um servidor caia ou fique fora do ar ou da rede, o sistema de arquivos deve manter as informações e ficar disponível de forma total ou parcialmente. Além disso, os usuários não necessitam saber como isso foi implementado e nem como funciona.

2.2.5 Consistência

Um sistema de arquivos distribuídos deve garantir a consistência dos arquivos de seus usuários. Quando um arquivo é replicado ou recuperado do cache de uma estação cliente, por exemplo, podem ocorrer demoras inevitáveis na propagação das modificações devido a latências de rede, podendo resultar na geração de inconsistências ou até perda das informações caso o sistema seja frágil a esse tipo de cenário (COULOURIS, 2005; SILBERSCHATZ, 2000).

2.2.6 Segurança

A finalidade da camada de segurança em sistemas distribuídos é de restringir o acesso à informação e aos recursos para usuários e processos não autorizados, fortalecendo a privacidade de indivíduos e organizações. Muitos dos recursos existentes nos sistemas distribuídos lidam com informações de elevado valor intrínseco para os usuários (COULOURIS, 2005), o que justifica a importância da segurança nestes sistemas.

2.2.7 Eficiência

Um sistema de arquivos distribuído deve fornecer um serviço que seja comparável (ou melhor) aos sistemas de arquivos locais, em termos de desempenho e confiabilidade. Para isso foram desenvolvidas técnicas para contornar os atrasos ocorridos com a comunicação entre os nodos, entre elas destacam-se o cache, que acessa a rede o mínimo possível, armazenando as informações acessadas constantemente na memória e também o balanceamento de carga, que graças a existência de replicas dos arquivos espalhadas em diversos servidores, pode optar por requisitar um arquivo de um ou mais servidores que estejam menos sobrecarregados, ganhando assim em tempo de processamento (COULOURIS, 2005).

2.2.8 Transparência

A transparência tem por objetivo esconder, sempre que possível, todos os detalhes do sistema que são irrelevantes do usuário (CHOW, 1998).

As técnicas de transparência de maior importância são a de acesso e localização, pois a presença ou ausência delas afeta fortemente a utilização dos recursos (COULOURIS, 2005). Como exemplo de uma transparência de acesso, pode-se considerar uma interface onde o usuário possa acessar seus arquivos e diretórios locais e remotos, enquanto que um exemplo da falta da transparência de acesso seria um sistema distribuído que não permitisse o acesso aos arquivos em um computador remoto a menos que o cliente faça uso, por exemplo, de um aplicativo de FTP (*File Transfer Protocol*) (COULOURIS, 2005).

3. FERRAMENTA DE SISTEMA DE ARQUIVOS DISTRIBUÍDOS

3.1 Distributed File System (DFS)

O *Distributed File System* faz com que múltiplos usuários compartilhem arquivos e recursos tendo como objetivo gerenciar recursos de forma eficiente mantendo-os disponíveis e seguros para os usuários, sua principal característica é a capacidade de gerenciar vários dispositivos de armazenamento, que se encontram dispersos em diferentes localidades.

Com a ferramenta, arquivos distribuídos a diversos servidores de um domínio podem ser acessados de forma transparente podendo ser conectados a um ou mais namespaces DFS, simulando que todo o material reside em um único servidor. Isso elimina a necessidade de os usuários acessarem vários locais da rede para localização de que precisam. A *namespace* DFS é uma exibição virtual das pastas compartilhadas em uma organização (MICROSOFT, 2013).

O DFS permite que administradores do sistema facilitem para os usuários o acesso e o gerenciamento de arquivos distribuídos fisicamente em uma rede. Com o DFS, você pode fazer com que arquivos distribuídos em vários servidores pareçam residir em um local na rede. Os usuários não precisam mais conhecer e especificar o local físico real dos arquivos para acessá-los. Quando um usuário exibe o *namespace*, as pastas parecem residir em um único disco rígido de alta capacidade. Os usuários podem navegar no *namespace* sem precisar saber os nomes dos servidores ou pastas compartilhadas que hospedam os dados. DFS também oferece muitos outros benefícios, incluindo tolerância a falhas e recursos de compartilhamento de carga, recursos que o tornam ideal para organizações que necessitam de um alto nível de disponibilidade a seus arquivos de dados (MICROSOFT, 2013).

4. METODOLOGIA

Após toda pesquisa exploratória, o trabalho implementado utilizou um equipamento notebook do fabricante Acer com processador Intel Core i3, 6gb de memória RAM DDR3, disco rígido de 500 Gb, com Windows Seven Professional 64 bits como Sistema Operacional.

Para demonstrar cada servidor virtualizado e a implementação do trabalho, foi criado um usuário e dois servidores Windows, cuja máquina virtual que será utilizada pelo usuário foi de configuração Windows XP, gerenciados pelo software Oracle VM VirtualBox 4.2.8-83876, cada servidor virtualizado foi instalado o Microsoft Windows Server 2008 R2 Enterprise.

No VM VirtualBox foi escolhida a rede interna entre as máquinas virtuais, não havendo interação com a máquina física. Todas as máquinas tiveram seu IP fixo.

Foi configurado o DFS em ambos servidores virtualizados, onde podemos mostrar a replicação dos arquivos criados pelo usuário do Windows XP fazendo com que os arquivos pudessem ser acessados e salvos de forma segura, transparente, consistente e tolerante a falhas.

Com o ambiente preparado, o objetivo desse trabalho era simular um cenário onde o usuário tem acesso ao domínio, podendo criar, salvar, excluir, modificar os arquivos mesmo com o desligamento de um dos servidores, mostrando a eficiência do sistema de arquivos distribuídos e fazendo com que o usuário não perceba o que tenha acontecido.

5. RESULTADOS

5.1 Implementação do DFS

Para a criação do ambiente foi preciso à instalação e configuração dos sistemas e rede abaixo conforme figura 1:

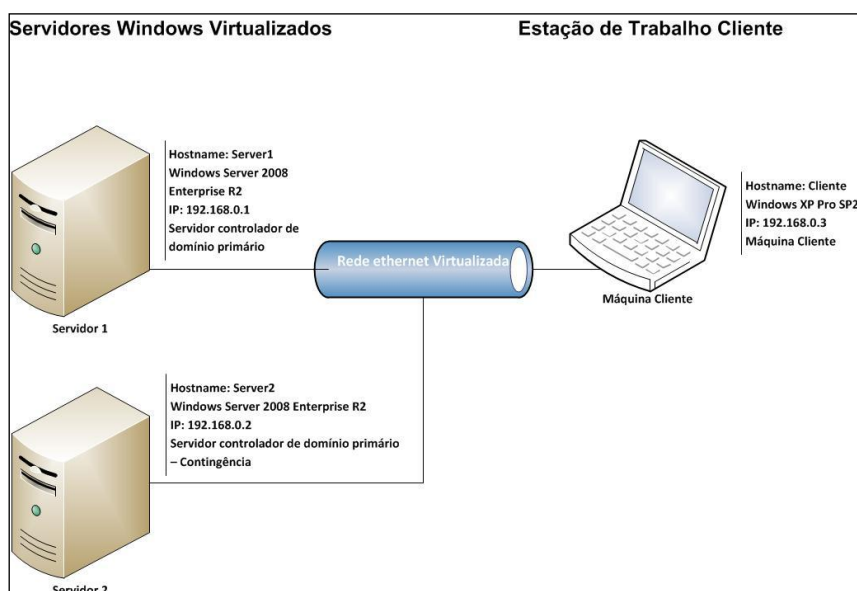


Figura 1 Ambiente virtualizado

Após a instalação do *Windows Server*, o DFS teve que ser habilitado em ambos servidores. Com o “*Server Manager*” aberto tivemos que adicionar e habilitar a função “*File Services*” e marcar os módulos do DFS (*Namespace e Replication*).

Para que a transparência de localização fosse possível, foi habilitada também a função do *Active Directory Domain Services* (ADDS), para criação do domínio da rede, transformando o SERVER1 em um controlador de domínio primário e o SERVER2 em controlador de domínio backup, e após a criação do domínio: “tcc.local”, o usuário acessa as pastas sem que consiga identificar a localização física do arquivo. Ou seja, o usuário irá acessar o domínio onde pode criar os arquivos, modificar, excluir e poderá acessar outras pastas disponíveis a ele. As configurações foram feitas em ambos os servidores, caso o servidor primário torne-se indisponível.

E então criar um novo *namespace* do DFS dentro da guia “*DFS Management*”. O *namespace* permite que agrupe pastas compartilhadas localizadas em diferentes servidores em um ou mais namespaces estruturados logicamente. Cada *namespace* aparece aos usuários como uma única pasta compartilhada com uma série de subpastas. Entretanto, a estrutura subjacente do *namespace* pode consistir de várias pastas compartilhadas que se localizam em diferentes servidores e em vários lugares.

Foi configurado então um novo Grupo de Replicação do DFS para os diretórios do *Namespace* criados. A Replicação DFS usa muitos processos sofisticados para manter os dados sincronizados em vários servidores. Esta configuração também é realizada no “*DFS Management*” através de um assistente, onde podemos visualizar na figura 9 a configuração de Replicação do DFS. Podem-se ter diversas configurações disponíveis tais como os servidores membros do grupo (no mínimo dois servidores), a topologia com que será configurada a replicação, largura de banda utilizada, dias e horários específicos para a replicação, membro primário (que terá seus dados replicados para outros servidores) e por fim o diretório local que receberá a replicação física dos arquivos no membro secundário de replicação. Com todas as características necessárias para o funcionamento do DFS e após todas as configurações serem um sucesso, nosso ambiente esta criado para a realização dos testes.

5.2 Transparência de acesso

Para que a que a transparência de acesso aconteça, há a necessidade de que o usuário saiba o domínio que irá acessar em nosso caso o usuário acessa via mapeamento de pasta, o usuário não consegue identificar a localização física de onde seus arquivos estão sendo salvos. O diretório mapeado foi o diretório virtual criado anteriormente pelo *Namespace*, que concede acesso às pastas do DFS, o usuário terá a impressão de que seus arquivos estão em seu computador.

Com a criação do mapeamento foi possível concluir que nosso ambiente preparado teve o objetivo alcançado, onde o usuário conseguia acessar, criar, modificar, excluir perfeitamente seus arquivos do DFS, como qualquer outra pasta local de seu computador.

5.3 Transparência de replicação e localização

Foi testada a transparência de replicação e localização, a partir de que um documento era criado, o usuário não saberia sua localização e não saberia que existiam cópias em outro servidor, servindo assim de *backup*. Em ambos os servidores foram configurados e ativados a replicação de DFS.

É possível verificar que para a criação do arquivo nos servidores é necessário aguardar alguns segundos, após isso a replicação estava concluída, ao simular a indisponibilidade do servidor principal, SERVER1, percebe-se que a máquina cliente fica lenta conforme o servidor backup, SERVER2, assumia como contingência, porém após alguns instantes

não havia mais oscilação e o usuário ainda conseguia acessar seus arquivos, modificar, excluir, sem qualquer problema.

Mesmo após a indisponibilidade do servidor e pelo fato de que o usuário conseguia acessar os arquivos disponíveis no DFS como se fosse um arquivo em seu computador e desconheça a existência de réplicas de seus arquivos em outros servidores, podemos dizer que este teste foi considerado bem sucedido.

6 CONSIDERAÇÕES FINAIS

O desenvolvimento na área de tecnologia da informação continua intenso e aumentando significativamente, e uma prova disso é o crescente aumento da rede de computadores nas empresas, a necessidade de sistemas de arquivos distribuídos vem crescendo, o que contribui significativamente com a organização e controle dos arquivos das empresas. Atualmente é indispensável que as empresas possuam sistemas que garantam a segurança e disponibilidade das informações sempre que preciso.

O trabalho acima desenvolvido teve como finalidade demonstrar a funcionalidade do DFS no ambiente Windows Server 2008 e sua importância, exemplificando uma situação de indisponibilidade do sistema, onde foi desligado o servidor controlador de domínio primário para que assim o servidor controlador de domínio backup pudesse assumir sua função, e mostrando que através do sistema de arquivos distribuídos o usuário poderá acessar seus arquivos mesmo com a falha, o que torna um ambiente em que trabalha mais seguro, estável e confiável para uma empresa, pois a perda de informações devido falhas no sistema pode causar grandes problemas às organizações, o que não pode ser permitido em um mundo cada vez mais competitivo, onde a tecnologia e sucesso caminham juntos.

Segundo a bibliografia pesquisada de Coulouris (2005); Tanenbaum (2000) e Silberschatz (2000) os sistemas de arquivos são projetados para armazenar e gerenciar um grande número de arquivos, com recursos para criação, atribuição de nomes e exclusão de arquivos. E os testes realizados foram importantes, pois através deles foi possível verificar o comportamento prático do DFS, desde sua instalação e configuração, onde o sistema de arquivo distribuído se mostrou muito estável e de fácil manuseio. A pesquisa tanto bibliográfica quanto pratica foi de extrema importância para a compreensão deste trabalho, pois possibilitou atingir os objetivos e demonstrar a importância e veracidade de sistemas de arquivos distribuídos nas organizações.

REFERÊNCIAS

- CHOW, Randy; JOHNSON, Theodore. **Distributed operating systems and algorithms. Reading**, Massachusetts: Addison Wesley Longman, 1998. 569 p.
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Distributed systems: concepts and design**. 4. ed. Harlow: Addison Wesley Longman, 2005. 927 p.
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas Distribuídos: Conceitos e Projeto**. 4. ed. Bookman, 2007. 784 p.
- GALLO, Michael A.; HANCOCK, William M. **Comunicação Entre Computadores e Tecnologias de Rede**. Brasil: Thomson, 2003. 673 p.
- GIL, Antonio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2008.

- KON, Fabio. **Sistemas de Arquivos Distribuídos**. São Paulo: 1994. 154 p. Tese (Mestrado em Matemática Aplicada) - Instituto de Matemática e Estatística, Universidade de São Paulo, 1994.
- MACHADO, Berenger; MAIA, Luiz Paulo. **Arquitetura de sistemas operacionais**. Rio de Janeiro : Livros Técnicos e Científicos Editora, 1997.
- MICROSOFT. **Como DFS funciona**. Disponível em: <
http://technet.microsoft.com/en-us/library/cc782417.aspx&usg=ALkJrhjNG_GCBh6fO4ITd19xL18wJ5wH3Q>.
Acesso em: 16 maio 2013.
- MICROSOFT. **O que é o DFS?**. Disponível em: < <http://technet.microsoft.com/en-us/library/cc779627.aspx&usg=ALkJrhgqOtxNSI7g6lO3KdfQfCJjWuDENA>>.
Acesso em: 16 maio 2013.
- MICROSOFT. **Visão geral sobre o Sistema de Arquivos Distribuídos**. Disponível em: <[http://technet.microsoft.com/pt-br/library/cc738688\(v=ws.10\).aspx](http://technet.microsoft.com/pt-br/library/cc738688(v=ws.10).aspx)>. Acesso em: 16 maio 2013
- SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. **Applied Operating System Concepts**. New York: John Wiley & Sons, 2000. 840 p.
- SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. **Applied Operating System Concepts**. New York: John Wiley & Sons, 2000. 840 p.
- TANENBAUM, Andrew S.; STEEN, Mararte Van. **Sistemas Distribuídos: princípios e paradigmas**. 2. ed. Prentice-hall, 2007. 416 p.
- TANENBAUM, Andrew S. **Computer Networks**. New Jersey: Prentice Hall, 2000. 384 p.
- TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Rio de Janeiro : Prentice-Hall do Brasil, 1995. 493 p.