



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Câmpus de São José do Rio Preto

Lúcio Rodrigo de Carvalho

Disponibilidade em um sistema de arquivos distribuído
flexível e adaptável

São José do Rio Preto
2014

Lúcio Rodrigo de Carvalho

Disponibilidade em um sistema de arquivos distribuído
flexível e adaptável

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientadora: Prof^ª. Dr^ª. Renata Spolon Lobato
Co-orientador: Prof. Dr. Aleardo Manacero Junior

São José do Rio Preto
2014

Carvalho, Lúcio Rodrigo de.

Disponibilidade em um sistema de arquivo distribuído flexível e adaptável / Lúcio Rodrigo de Carvalho. -- São José do Rio Preto, 2014

115 f. : il., gráfs., tabs.

Orientador: Renata Spolon Lobato

Coorientador: Aleardo Manacero Junior

Dissertação (mestrado) – Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Sistemas de arquivos distribuídos.
3. Tolerância a falha (Computação) I. Lobato, Renata Spolon.
II. Manacero Junior, Aleardo. III. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas.
IV. Título.

CDU – 681.3

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Lúcio Rodrigo de Carvalho

Disponibilidade em um sistema de arquivos distribuído flexível e adaptável

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Comissão Examinadora

Prof^ª. Dr^ª. Renata Spolon Lobato
UNESP – São José do Rio Preto
Orientadora

Prof. Dr. Norian Marranghello
UNESP – São José do Rio Preto

Prof^ª. Dr^ª. Sarita Mazzini Bruschi
USP – São Carlos

São José do Rio Preto
04 de Dezembro de 2014

DEDICATÓRIA

Dedico este trabalho ao Divino Pai Eterno, a São Francisco de Assis e a São João Paulo II, sempre presentes em minhas orações e que estiveram comigo em todos os momentos desta etapa.

AGRADECIMENTOS

A Deus.

Um agradecimento especial à minha esposa, Luciana, por estar sempre presente nos momentos que precisei durante esta etapa.

Ao Instituto Federal de São Paulo pelo incentivo e apoio.

À professora Doutora Renata Spolon Lobato e ao professor Doutor Aleardo Manacero Junior pela orientação.

Aos professores Mestre Daniel Corrêa Lobato, Mestre Júlio Fernando Lieira, Doutor Osvaldo Severino Junior, Doutor Paulo César Mioralli, Doutor Márcio Andrey Teixeira, Doutor Rodolfo Meneguette Ipólito, Mestre Eros Schettini Roman e Doutor Rinaldo Macedo de Moraes, amigos de trabalho e incentivadores.

Aos professores Doutor Fernando Ferrari, Doutor Norian Marranghello, Doutora Sarita Mazzini Bruschi e Doutor Aledir Silveira Pereira.

Às professoras Beca e Doutora Silvia Jorge de Almeida Martins.

Ao Mestre Silas Evandro Nachif Fernandes.

Aos colegas que fiz na Unesp: todo o pessoal do GSPD, Mestre Denison, Gabriel Saraiva, Gabriel Covello, Diogo Tavares, Cássio Forte e, em especial ao Danilo Segura, Thiago Okada e Matheus Della Croce: por tudo o que fizemos sempre um ajudando ao outro.

Às bibliotecárias Luciane Passoni e Milena.

À Fapesp pelos equipamentos utilizados neste trabalho sob o processo número 2012/02926-5.

Muitíssimo obrigado a todos por participarem deste momento importante da minha carreira profissional.

Algo novo me espera...

"Ninguém pode voltar atrás e fazer um novo começo. Mas qualquer um pode recomeçar e fazer um novo fim".

Chico Xavier

RESUMO

Um sistema de arquivos distribuído permite que usuários e aplicações possam armazenar e compartilhar dados, acessando tais recursos remotamente como se fossem locais. As características de um sistema de arquivos distribuído podem ser variadas. Assim, é impossível conceber um sistema abrangendo todas as características desejáveis, tais como: transparência, desempenho, escalabilidade, confiabilidade e disponibilidade, por exemplo. O sistema de arquivos distribuído Flexível e Adaptável (FlexA) incorpora importantes características do NFS, AFS, GFS e Tahoe-LAFS. Este sistema elimina a necessidade de um servidor principal (como o *master* no GFS ou o *Introducer* no Tahoe-LAFS). Os arquivos são armazenados em dois grupos de servidores: um grupo de leitura, onde somente dados são armazenados, e um grupo de escrita, onde dados e *metadados* são armazenados. A disponibilidade é provida por um mecanismo semelhante ao apresentado pelo Tahoe-LAFS. No presente estudo são apresentadas as melhorias alcançadas por meio da disponibilidade do FlexA. Os detalhes sobre as modificações no FlexA, bem como os resultados obtidos indicam que o FlexA é uma importante opção de sistema de arquivos distribuído.

Palavras-chave: sistema de arquivos distribuído; tolerância à falhas; disponibilidade.

ABSTRACT

A distributed file system allows users and applications to store and share data, accessing such resources remotely as if they were local. The characteristics of a distributed file system can be varied. Thereby, it is impossible to design a system covering all desirable characteristics, such as transparency, performance, scalability, reliability and availability, for example. The Flexible and Adaptable distributed file system (FlexA) incorporate important characteristics of NFS, AFS, GFS and Tahoe-LAFS. It eliminates the need for main server (such as the master in GFS or the introducer in Tahoe-LAFS). File storage is provided by two group of server: a reading group, where only data is found, and a writing group, where data and metadata are stored. Availability is provided in FlexA through a mechanism similar to the one presented by Tahoe-LAFS. In the present study the improvements achieved through the availability and performance of FlexA are presented. Details about the changes in the FlexA as well as the obtained results indicate that the FlexA is an important option for distributed file system.

Keywords: distributed file system; fault-tolerance; availability.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo cliente servidor	27
Figura 2 - (a) Modelo de acesso remoto e (b) Modelo carga atualização	28
Figura 3 - Serviços oferecidos a clientes por servidores	28
Figura 4 - Distribuição de arquivos	29
Figura 5 - Aplicativo baseado em arquitetura <i>peer-to-peer</i>	30
Figura 6 - Arquitetura do NFS	32
Figura 7 - Distribuição de processos no AFS	34
Figura 8 - Arquitetura do <i>Google File System</i>	36
Figura 9 - Arquitetura do Lustre	38
Figura 10 - Interação entre os elementos essenciais do Tahoe-LAFS	39
Figura 11 - Visão geral do Tahoe-LAFS	40
Figura 12 - Arquitetura do FlexA original	42
Figura 13 - Módulos do FlexA original	43
Figura 14 – Leitura e escrita no FlexA original	45
Figura 15 - Arquitetura do FlexA desenvolvido	49
Figura 16 - Módulos do FlexA desenvolvido	50
Figura 17 - Equação para coleta de métricas	52
Figura 18 - Operação de escrita no FlexA desenvolvido	53
Figura 19 - Operação de leitura no FlexA desenvolvido	55
Figura 20 – Servidores primários esperados, servidores ativos e desativados	56
Figura 21 - Fase ‘detecção’ na avaliação de servidores primários	58
Figura 22 - Fases ‘eleição’ e ‘substituição’ na avaliação de servidores primários	59
Figura 23 - Índice de disponibilidade na sobrecarga do servidor primário	61
Figura 24 - Classificação na sobrecarga do servidor primário	61
Figura 25 - Avaliação da sobrecarga do servidor primário	63
Figura 26 - Índice de disponibilidade na sobrecarga do servidor secundário	65
Figura 27 - Lista classificatória com classificações diferentes	66
Figura 28 - Avaliação da sobrecarga do servidor secundário	67
Figura 29 – Informações de configuração no cliente	70
Figura 30 – Início da autoavaliação no servidor primário	71

Figura 31 – Operação de escrita no cliente – envio de porções	72
Figura 32 – Operação de escrita no cliente – transferência do arquivo.....	72
Figura 33 - Operação de leitura no cliente – verificação de permissões	73
Figura 34 – Solicitação das porções aos servidores	73
Figura 35 - Envio de porção de um arquivo a partir de servidor	74
Figura 36 – Avaliação da sobrecarga de servidor primário	74
Figura 37 - Servidor primário sobrecarregado	75
Figura 38 - Solicitação e devolução de métricas do servidor secundário	75
Figura 39 - Servidor secundário inicia Coletor - servidor primário	75
Figura 40 - Autoavaliação da sobrecarga do servidor secundário	76
Figura 41 - Parando autoavaliação do servidor secundário	76
Figura 42 - Cliente inicia Coletor - servidor secundário	77
Figura 43 - Cenário de avaliação.....	86

LISTA DE TABELAS

Tabela 1 – Métricas para compor índice de disponibilidade do servidor primário	60
Tabela 2 – Métricas para compor índice de disponibilidade do servidor secundário.....	65
Tabela 3 – Resultado da avaliação de sondagem de servidores primários	77
Tabela 4 – Eleição de servidores secundários	78
Tabela 5 – Tempos médios das etapas de sondagem, eleição e sincronização	78
Tabela 6 – Informações de sobrecarga do servidor primário	82
Tabela 7 – Servidores secundários no teste de sobrecarga dos servidores primários.....	82
Tabela 8 – Informações de sobrecarga do servidor secundário.....	83
Tabela 9 – Clientes na sobrecarga do servidor secundário.....	84
Tabela 10 – Consumo de memória na sobrecarga primário (escrita)	110
Tabela 11 – Consumo de memória na sobrecarga primário (leitura)	110
Tabela 12 – Atividade em disco na sobrecarga do servidor primário (escrita).....	111
Tabela 13 – Atividade em disco na sobrecarga do servidor primário (leitura)	111
Tabela 14 – Tempos de escrita no FlexA original.....	111
Tabela 15 – Tempos de leitura no FlexA original.....	112
Tabela 16 – Tempos de escrita no Tahoe-LAFS	112
Tabela 17 – Tempos de leitura no Tahoe-LAFS	113
Tabela 18 – Tempos de escrita no NFS.....	113
Tabela 19 – Tempos de leitura no NFS	114
Tabela 20 – Tempos de escrita no FlexA desenvolvido.....	114
Tabela 21 – Tempos de leitura no FlexA desenvolvido	115
Tabela 22 – Comparação de tempos na escrita e leitura por 1 cliente.....	115
Tabela 23 – Comparação de tempos na escrita e leitura por 2 clientes	116
Tabela 24 – Comparação de tempos na escrita e leitura por 4 clientes.....	116
Tabela 25 – Comparação de tempos na escrita e leitura por 8 clientes.....	116
Tabela 26 – Comparação de tempos na escrita e leitura por 16 clientes.....	117

LISTA DE GRÁFICOS

Gráfico 1 – Consumo de memória na operação de escrita	79
Gráfico 2 – Atividade em disco na operação de escrita	80
Gráfico 3 – Consumo de memória na operação de leitura	80
Gráfico 4 – Atividade em disco na operação de leitura	81
Gráfico 5 – Tempos de escrita no FlexA original	88
Gráfico 6 – Tempos de leitura no FlexA original	89
Gráfico 7 – Tempos de escrita no Tahoe-LAFS	90
Gráfico 8 – Tempos de leitura no Tahoe-LAFS	90
Gráfico 9 – Tempos de escrita no NFS	91
Gráfico 10 – Tempos de leitura no NFS	92
Gráfico 11 – Tempos de escrita no FlexA desenvolvido	93
Gráfico 12 – Tempos de leitura no FlexA desenvolvido	94
Gráfico 13 – Comparação do tempo de escrita para 1 cliente	95
Gráfico 14 – Comparação do tempo de leitura para 1 cliente	96
Gráfico 15 – Comparação do tempo na escrita de 8 clientes	97
Gráfico 16 – Comparação do tempo na leitura de 8 clientes	98
Gráfico 17 – Comparação do tempo na escrita de 16 clientes	99
Gráfico 18 – Comparação do tempo na leitura de 16 clientes	100

LISTA DE QUADROS

Quadro 1 – Comparação entre o FlexA original e o FlexA desenvolvido	48
Quadro 2 – Comandos do FlexA desenvolvido.....	71
Quadro 3 – Disposição dos servidores na avaliação.....	86

LISTA DE ABREVIATURAS E SIGLAS

AFS	<i>Andrew File System</i>
CMU	<i>Carnegie Mellon Univerity</i>
FID	<i>File Identifier</i>
FlexA	Sistema de arquivos distribuído Flexível e Adaptável
FTP	<i>File Transfer Protocol</i>
GFS	<i>Google File System</i>
HTTP	<i>HyperText Transfer Protocol</i>
IBM	<i>International Business Machines</i>
IP	<i>Internet Protocol</i>
MDS	<i>Meta-data Services</i>
NFS	<i>Network File System</i>
NTP	<i>Network Time Protocol</i>
OSC	<i>Object Storage Client</i>
OSS	<i>Object Storage Servers</i>
OST	<i>Object Storage Target</i>
RPC	<i>Remote Procedure Call</i>
SAD	Sistema de Arquivos Distribuído
SSL	<i>Security Soccer Layer</i>
TCP	<i>Transmission Control Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
UDP	<i>User Datagram Protocol</i>
UUID	<i>Universally Unique Identifier</i>
VFS	<i>Virtual File System</i>
WAN	<i>Wide Area Network</i>

SUMÁRIO

LISTA DE ILUSTRAÇÕES	i
LISTA DE TABELAS	iii
LISTA DE GRÁFICOS	iv
LISTA DE QUADROS	v
LISTA DE ABREVIATURAS E SIGLAS	vi
1 INTRODUÇÃO	
1.1 Motivação	19
1.2 Objetivos	20
1.3 Organização do texto	21
2 SISTEMAS DE ARQUIVOS DISTRIBUÍDOS	
2.1 Considerações iniciais	22
2.2 Características dos sistemas de arquivos distribuídos	23
2.2.1 Transparência	23
2.2.2 Replicação	23
2.2.3 Tolerância à falhas	24
2.2.4 Consistência	24
2.2.5 Segurança	25
2.2.6 Escalabilidade	25
2.2.7 Disponibilidade	25
2.2.8 Sincronização	26
2.3 Modelos Arquiteturais	26
2.3.1 Arquitetura cliente-servidor	27
2.3.2 Arquitetura baseada em <i>cluster</i>	28
2.3.3 Arquitetura <i>peer-to-peer</i>	29
2.4 Considerações Finais	30
3 ESTUDOS DE CASOS	
3.1 Considerações iniciais	31

3.2 NFS – <i>Network File System</i>	31
3.2.1 Arquitetura.....	31
3.2.2 <i>Cache</i> e segurança	32
3.3 AFS – <i>Andrew File System</i>	33
3.3.1 Arquitetura.....	33
3.3.2 <i>Cache</i> e segurança	34
3.4 GFS – <i>Google File System</i>	35
3.4.1 Arquitetura.....	35
3.4.2 Disponibilidade.....	37
3.5 Lustre	37
3.5.1 Arquitetura.....	37
3.6 Tahoe-LAFS	38
3.6.1 Arquitetura.....	39
3.6.2 Disponibilidade.....	40
3.7 FlexA original	41
3.7.1 Arquitetura.....	41
3.7.2 Módulos.....	43
3.7.3 Flexibilidade	44
3.7.4 Utilização de <i>hardware</i> de baixo custo	44
3.7.5 Desempenho	45
3.7.6 Controle de acesso e disponibilidade.....	45
3.8 Considerações finais	46
4 TRABALHO DESENVOLVIDO	
4.1 Arquitetura	49
4.2 Operação de escrita	51
4.3 Operação de leitura	53
4.4 Autoavaliação do servidor primário	55
4.5 Avaliação da sobrecarga do servidor primário	60
4.6 Avaliação da sobrecarga do servidor secundário	64
4.7 Considerações finais	68
5 VALIDAÇÃO DO SISTEMA	
5.1 Considerações iniciais	69

5.2 Cenário para validação	69
5.3 Funcionamento do sistema	70
5.3.1 Operação de escrita	71
5.3.2 Operação de leitura	73
5.3.3 Sobrecarga do servidor primário	74
5.3.4 Sobrecarga do servidor secundário	76
5.4 Validação da autoavaliação do servidor primário	77
5.5 Validação da avaliação da sobrecarga do servidor primário	78
5.6 Validação da avaliação da sobrecarga do servidor secundário	83
5.7 Considerações finais	84
6 AVALIAÇÃO E COMPARAÇÃO DE DESEMPENHO	
6.1 Considerações iniciais	85
6.2 Cenário de avaliação	85
6.3 Avaliação de desempenho	87
6.3.1 FlexA original	88
6.3.2 Tahoe-LAFS	89
6.3.3 NFS – <i>Network File System</i>	91
6.3.4 FlexA desenvolvido	93
6.4 Comparação de desempenho	95
6.5 Considerações finais	100
7 CONCLUSÃO	102
REFERÊNCIAS	104
APÊNDICE A – Instalação do FlexA Desenvolvido	108
APÊNDICE B – Tabelas de tempo	110

1 INTRODUÇÃO

1.1 Motivação

A informática tem crescido a passos largos desde a popularização do computador pessoal, fazendo com que novos serviços passassem a ser agregados nesta área. Aliada ao oferecimento desses serviços surge a necessidade de armazenamento de dados os quais, inicialmente, eram armazenados em servidores isolados.

O grande problema em se oferecer dados a partir de servidores isolados (centralizados) é a questão de que falhas físicas ou lógicas também podem ocorrer que vai desde uma interrupção na rede, a falta de energia ou até problemas relacionados a processos no servidor. Por outro lado, passou-se a pensar na descentralização do armazenamento de dados e serviços a partir de um sistema distribuído (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Segundo Coulouris, Dollimore e Kindberg (2007), um sistema distribuído pode ser definido como sendo aquele nos quais os componentes de *hardware* ou *software*, localizados em computadores interligados em rede, se comunicam e coordenam suas ações pela troca de mensagens. Nessa linha, Tanenbaum e Steen (2007) caracterizam um sistema distribuído como um conjunto de computadores independentes que se apresenta aos seus usuários como sendo um sistema único e coerente.

A principal motivação para a concepção de um sistema distribuído está ligada ao compartilhamento de recursos que vão desde componentes de *hardware* (discos ou impressoras) até um banco de dados (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Como um exemplo típico de sistema distribuído, Coulouris, Dollimore e Kindberg (2007) citam a Internet, uma WAN (*Wide Area Network*) composta por um conjunto de redes de computadores de vários tipos, que estão interligados. Para que a comunicação possa ocorrer entre os computadores que fazem parte dessa grande rede, os computadores interagem pela troca de mensagens, viabilizando a utilização de serviços tais como troca de arquivos e mensagens eletrônicas.

Os sistemas distribuídos também podem ser utilizados para tarefas de computação de alto desempenho. Como exemplo, Tanenbaum e Steen (2007) citam a computação em *cluster*.

Os sistemas de arquivos distribuídos surgiram com base em uma das motivações principais para a adoção de um sistema distribuído: o compartilhamento de recursos.

Um sistema de arquivos distribuído permite que usuários e aplicações possam armazenar e compartilhar dados, acessando tais recursos remotamente como se fossem locais (BZOCH; SAFARIK, 2011b).

Tanenbaum e Steen (2007) afirmam que os primeiros sistemas de arquivos distribuídos foram desenvolvidos nos anos 70, sendo que o *Sun Network File System* (NFS) se tornou disponível no início dos anos 80.

No início do desenvolvimento dos sistemas de arquivos distribuídos, foram detectados muitos requisitos que poderiam fazer com que esses sistemas pudessem funcionar de uma forma melhorada e segura. Os primeiros sistemas ofereciam requisitos ligados a transparência de acesso e transparência de localização; no entanto, atualmente tais sistemas necessitam de mais requisitos para que possam operar de uma forma melhorada e segura, tais como transparência, replicação, tolerância a falhas, consistência, escalabilidade, disponibilidade e sincronização.

Como exemplos de sistemas de arquivos distribuídos citam-se o *Andrew File System*, o GFS (sistema desenvolvido pela Google no sentido de suprir suas necessidades devido ao fato de lidar com arquivos na ordem de *gigabytes*), o Lustre, o Tahoe-LAFS e o FlexA, foco deste trabalho. O FlexA é uma opção eficiente de sistema de arquivos distribuído, auxiliando o usuário no armazenamento de arquivos; agrega ainda características de um sistema deste tipo, tais como transparência, tolerância a falhas, segurança e facilidade de uso.

1.2 Objetivos

Este trabalho aborda a melhoria no que diz respeito à disponibilidade do sistema FlexA. O FlexA é um sistema de arquivos distribuído Flexível e Adaptável desenvolvido no GSPD (Grupo de Sistemas Paralelos e Distribuídos) da Unesp – São José do Rio Preto, escolhido devido ao fato de lidar com *hardware* de baixo custo e ser *open source*. Durante este trabalho, o FlexA é evidenciado como FlexA original para diferenciar da versão desenvolvida neste trabalho, chamada de FlexA desenvolvido.

Os melhoramentos estão relacionados à questão da disponibilidade deste sistema através do desenvolvimento de módulos que lidam com replicação, autoavaliação de servidores primários e verificação de sobrecarga dos servidores primários e servidores secundários.

1.3 Organização do texto

No capítulo 2 são apresentados os conceitos que envolvem os sistemas de arquivos distribuídos, assim como suas características e modelos arquiteturais.

No capítulo 3 são apresentados alguns sistemas de arquivos distribuídos existentes como estudos de casos.

No capítulo 4 é apresentado o trabalho desenvolvido, através das melhorias agregadas à disponibilidade do sistema FlexA original: o desenvolvimento de módulos que lidam com replicação, operação de autoavaliação de servidores primários e verificação de sobrecarga dos servidores primários e servidores secundários.

Nos capítulos 5 e 6 são apresentadas a validação do sistema e avaliações e comparações de desempenho dos sistemas de arquivos distribuídos Tahoe-LAFS, NFS, FlexA original e FlexA desenvolvido, respectivamente.

Por fim, no capítulo 7, é apresentada a conclusão deste trabalho e são discutidos desdobramentos do projeto.

2 SISTEMAS DE ARQUIVOS DISTRIBUÍDOS

2.1 Considerações iniciais

Os primeiros sistemas de arquivos foram originalmente criados para computadores *desktop*, centralizados através do oferecimento de uma interface para que os dados pudessem ser armazenados em disco.

Posteriormente, receberam o recurso de controle de acesso e proteção de arquivos para que pudessem ser mais seguros. Com o surgimento das redes e a necessidade de comunicação, os sistemas de arquivos distribuídos foram apresentados (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Neste sentido, Tanenbaum e Steen (2007) citam que os primeiros sistemas de arquivos distribuídos foram desenvolvidos na década de 70; a exemplo do NFS, disponibilizado para uso comercial em meados de 1980.

Um sistema de arquivos distribuído permite aos programas armazenarem e acessarem arquivos remotos exatamente como se fossem locais, permitindo que os usuários acessem arquivos a partir de qualquer computador em uma rede (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Segundo Bai e Wu (2011), um sistema de arquivos distribuído consiste em um servidor de *metadados* (responsável por gerenciar as informações do arquivo), um servidor de armazenamento (responsável por armazenar os dados) e clientes (usuários do sistema). No entanto, em alguns sistemas, um mesmo servidor pode fazer o papel de servidor de dados e servidor de *metadados*.

Cabe ressaltar que *metadados* de um arquivo correspondem às informações extras do arquivo, que são necessárias para que o sistema de arquivos distribuído possa gerenciar os arquivos (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Na concepção de um sistema de arquivos distribuído, algumas características devem ser consideradas, tais como: transparência, replicação, tolerância à falhas, consistência, segurança, escalabilidade, disponibilidade, eficiência e sincronização.

2.2 Características dos sistemas de arquivos distribuídos

O projeto de desenvolvimento de um sistema de arquivos distribuído está ligado a várias características. A seguir são apresentadas algumas dessas características.

2.2.1 Transparência

Em um sistema de arquivos distribuído, transparência é a capacidade do sistema de mover arquivos entre os componentes que fazem parte do sistema de forma que o usuário final perceba o sistema como um todo (BZOCH; SAFARIK, 2011b).

Nessa linha, Tanenbaum e Steen (2007) citam que a meta de um sistema de arquivos distribuído é ocultar o fato de que processos e recursos estão disponíveis em computadores diferentes, ou seja, que estão fisicamente distribuídos.

Na reafirmação de Coulouris, Dollimore e Kindberg (2007), transparência de um sistema de arquivos distribuído está relacionada à ocultação, junto ao usuário final ou ao programador de aplicativos, da separação dos componentes do sistema de arquivos distribuído de modo que o sistema possa ser percebido como um todo, ao invés de uma coleção de itens independentes. Dentre os tipos de transparência temos: transparência de acesso, de localização, de escalabilidade, de falhas, de concorrência e de replicação (PUTTER; ROOS, 1994).

2.2.2 Replicação

A replicação é a operação pela qual um arquivo original (chamado de réplica primária ou réplica *master*) é copiado para outros nós que fazem parte de um sistema de arquivos distribuído no intuito de melhorar seu desempenho, confiabilidade e tolerância a falha. Neste sentido, os dados frequentemente utilizados ou importantes podem ser armazenados em vários nós para que, em caso de falha, possam ser recuperados (BZOCH; SAFARIK, 2011b).

Com o uso da replicação é possível aumentar a proporção do tempo que um sistema está acessível, garantindo assim a sua disponibilidade (TANENBAUM; STEEN, 2007).

Nessa linha, Coulouris, Dollimore e Kindberg (2007) reafirmam que a replicação melhora a disponibilidade do sistema, bem como a tolerância a falhas. Isso se deve ao fato de que, em caso de falha de um servidor, o arquivo possa ser procurado em outro servidor que tenha o conteúdo replicado.

2.2.3 Tolerância à falhas

Devido ao fato de que um sistema de arquivos distribuído ser composto por vários computadores que estão interconectados é comum a presença de falhas de *software*, *hardware* ou de comunicação nestas estações. Dessa forma, a ausência no oferecimento de um serviço pelo sistema ou componente pertencente ao sistema é caracterizada como uma falha (TANENBAUM; STEEN, 2007).

Segundo White (2003), o grau de tolerância à falha é diferenciado para cada sistema, sendo que nenhum sistema pode verdadeiramente ser construído para suportar qualquer tipo de combinação a falhas; existem sempre algumas combinações de eventos de falhas que não são tratados.

Nesta linha, Tanenbaum e Steen (2007) citam que um sistema de arquivos distribuído pode conviver com as falhas, contando que tais falhas possam ser mascaradas, toleradas ou até mesmo que o sistema possa se recuperar na ocorrência de tais falhas.

Coulouris, Dollimore e Kindberg (2007) citam alguns tipos de falhas: falha por omissão, falha por temporização, falha arbitrária e falha por queda.

Em muitos casos os projetos de sistemas de arquivos distribuídos utilizam a técnica de replicação, no sentido de criar grupos que possam ser acionados quando da ocasião de uma falha (TANENBAUM; STEEN, 2007).

2.2.4 Consistência

Pelo fato de existirem cópias de um mesmo arquivo replicadas em diferentes nós do sistema, um sistema de arquivos distribuído deve tomar iniciativas para manter a consistência dos arquivos (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Como exemplo, cita-se um sistema cujo arquivo foi modificado. Neste caso, todos os nós que detenham cópia deste arquivo devem ter suas definições para este arquivo atualizadas.

2.2.5 Segurança

Nos sistemas de arquivos distribuídos a questão da segurança pode ser resolvida utilizando mecanismos para garantir a comunicação e o controle de acesso.

Com relação à segurança na comunicação, é comum a utilização da técnica de criptografia simétrica ou assimétrica (HARRINGTON; JENSEN, 2003).

O controle de acesso pode ser realizado com a utilização do sistema *Kerberos* como protocolo de autenticação, mantendo a integridade dos dados (BZOCH; SAFARIK, 2011a). Segundo Neuman e Theodore (1994), o *Kerberos* é um serviço de autenticação distribuída que permite a um processo (um cliente) prover sua identidade a uma aplicação ou servidor, conservando a integridade e a confidencialidade dos dados enviados entre o cliente e o servidor.

Como outro exemplo, o controle de acesso abordado pelo NFS faz com que os direitos de acesso dos clientes sejam verificados ao acessar um servidor para efetuar uma operação.

2.2.6 Escalabilidade

Um sistema pode ser considerado escalável se permanece eficiente quando existe um aumento significativo do número de recursos e de usuários, a exemplo da Internet (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Nesta linha, Bzoch e Safarik (2011b) citam que um sistema de arquivos distribuído que mantém a escalabilidade deve ser capaz de aumentar o número de nós pertencentes ao sistema para que, assim, possa atender um número maior de serviços e clientes mantendo-se eficiente.

2.2.7 Disponibilidade

A Associação Brasileira de Normas Técnicas (ABNT) regulamenta através da norma NBR 5462, a questão da disponibilidade como sendo a capacidade de um item executar suas

funções de forma adequada em determinado espaço de tempo (Associação Brasileira de Normas Técnicas, 1994).

Nessa linha, Coulouris, Dollimore e Kindberg (2007) evidenciam que, em um sistema de arquivos distribuído, a disponibilidade está ligada a proporção de tempo em que um sistema está pronto para utilização.

Segundo Bzoch e Safarik (2011b), a disponibilidade de um sistema faz com que, na falha de um ou mais nós, outros nós estejam aptos a prover a funcionalidade do sistema.

A disponibilidade de um sistema pode ser aprimorada através das técnicas de replicação e tolerância à falhas (COULOURIS; DOLLIMORE; KINDBERG, 2007).

2.2.8 Sincronização

A sincronização é um item obrigatório quando se fala de sistemas de arquivos distribuídos pelo fato dos arquivos estarem compartilhados.

Como exemplo, cita-se um mesmo arquivo aberto por dois clientes: quando a operação de leitura vem depois da operação de escrita, basta mostrar o valor atual; caso existam duas operações de escrita consecutivas, seguidas de uma operação de leitura, basta mostrar o último conteúdo (TANENBAUM; STEEN, 2007).

Coulouris, Dollimore e Kindberg (2007) citam que a forma como a precedência nas operações é tratada é denominada de semântica. No exemplo apresentado é adotada a semântica *Unix*, no qual o sistema impõe uma ordenação em tempo absoluto. Outras técnicas relacionadas é a semântica de sessão e técnica de travamento de arquivo. Dessa forma, através da adoção de uma semântica, a questão da sincronização no uso de arquivos compartilhados pode ser resolvida.

Apresentadas as principais características ligadas aos sistemas de arquivos distribuídos, a seguir são evidenciadas possíveis arquiteturas para este tipo de sistema.

2.3 Modelos arquiteturais

A arquitetura dos sistemas de arquivos distribuídos diz respeito à forma em que estes sistemas podem ser organizados. Na maioria dos projetos, o modelo cliente-servidor é

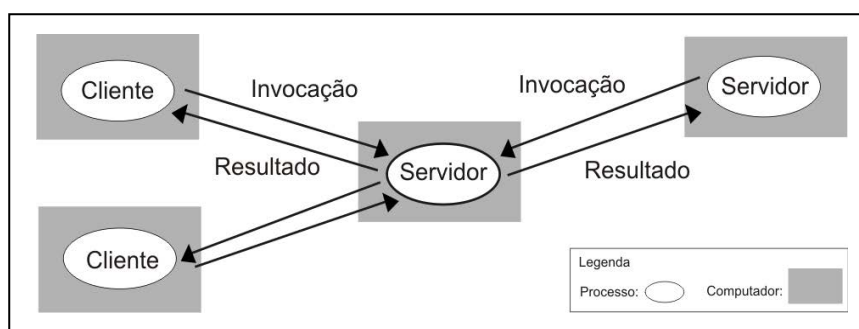
evidenciado; no entanto, existem outros modelos, tais como o modelo baseado em *cluster* e o modelo *peer-to-peer*.

2.3.1 Arquitetura cliente-servidor

Na arquitetura cliente-servidor, os clientes estão localizados em hospedeiros distintos, a partir dos quais podem acessar recursos compartilhados que são gerenciados pelos servidores, como apresentado na Figura 1.

Neste modelo, dependendo do caso, servidores podem tornar-se clientes de outros servidores; é o caso em que um servidor *web* pode ser cliente de um servidor que armazena os arquivos de páginas *web* que este primeiro está apto a exibir (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Figura 1 – Modelo cliente servidor



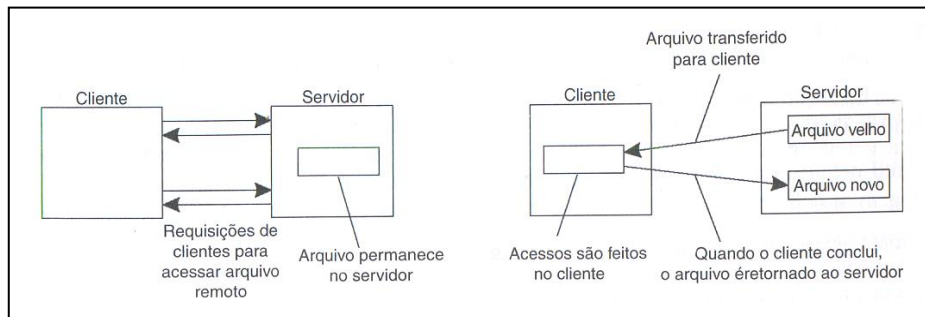
Fonte: Adaptado de Tanenbaum e Steen (2007)

Segundo Coulouris, Dollimore e Kindberg (2007), no modelo cliente-servidor cada servidor suporta um modelo de sistema compatível com o modelo de arquivos locais do cliente para que possa existir a compatibilidade entre ambos, sendo este de dois tipos: acesso remoto ou modelo carga/atualização.

No modelo de acesso remoto, o servidor oferece aos clientes acesso transparente através de uma interface a seu sistema de arquivos. Dessa forma, o cliente não sabe o local onde os arquivos estão armazenados. Essa abordagem também é conhecida como modelo de serviço de arquivo remoto (TANENBAUM; STEEN, 2007).

No modelo carga/atualização, o arquivo é transferido integralmente para o cliente para realizar modificações. Quando acaba de realizar as modificações, o arquivo é recarregado no servidor. Na Figura 2 são apresentados os modelos acesso remoto e carga/atualização.

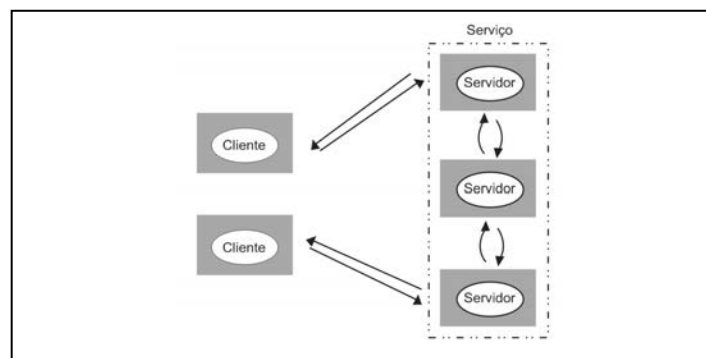
Figura 2 - (a) Modelo de acesso remoto e (b) Modelo carga atualização



Fonte: Adaptado de Tanenbaum e Steen (2007)

Uma variação do modelo cliente-servidor é aquele no qual serviços são oferecidos por vários servidores. Na Figura 3 são apresentados servidores oferecendo serviços a dois clientes distintos.

Figura 3 – Serviços oferecidos a clientes por servidores



Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2007)

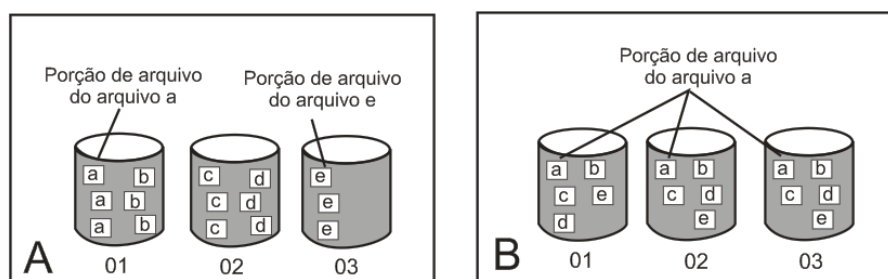
Neste caso, verifica-se a interação entre vários processos servidores em hospedeiros distintos para que um serviço possa ser oferecido ao cliente. Essa abordagem prevê a distribuição de arquivos entre os servidores envolvidos e a manutenção de réplicas visando a disponibilidade (COULOURIS; DOLLIMORE; KINDBERG, 2007).

2.3.2 Arquitetura baseada em *cluster*

Na arquitetura baseada em *cluster*, uma técnica conhecida é o desmembramento do arquivo em porções. Segundo Coulouris, Dollimore e Kindberg (2007), o desmembramento em porções é aconselhável quando o acesso paralelo no sistema de arquivos distribuído é eficiente.

Como exemplo, na operação de escrita (leia-se armazenamento no arquivo no sistema), um arquivo é desmembrado em porções e distribuído para vários servidores (entende-se por porção cada unidade resultante da divisão de um arquivo). Na operação de leitura, as porções são lidas de forma paralela a partir dos servidores onde foram armazenados, sendo apresentados aos clientes. Na Figura 4 é apresentada a distribuição de arquivos; no entanto, existem duas formas de lidar com a distribuição dos mesmos.

Figura 4 – Distribuição de arquivos



Fonte: Adaptado de Tanenbaum e Steen (2007)

Na Figura 4 (A) é apresentado o desmembramento do arquivo 'a' em três porções, tendo todas as porções deste arquivo armazenadas em um mesmo servidor (01). Da mesma forma, o arquivo 'e' é dividido em três porções e armazenado no mesmo servidor (03).

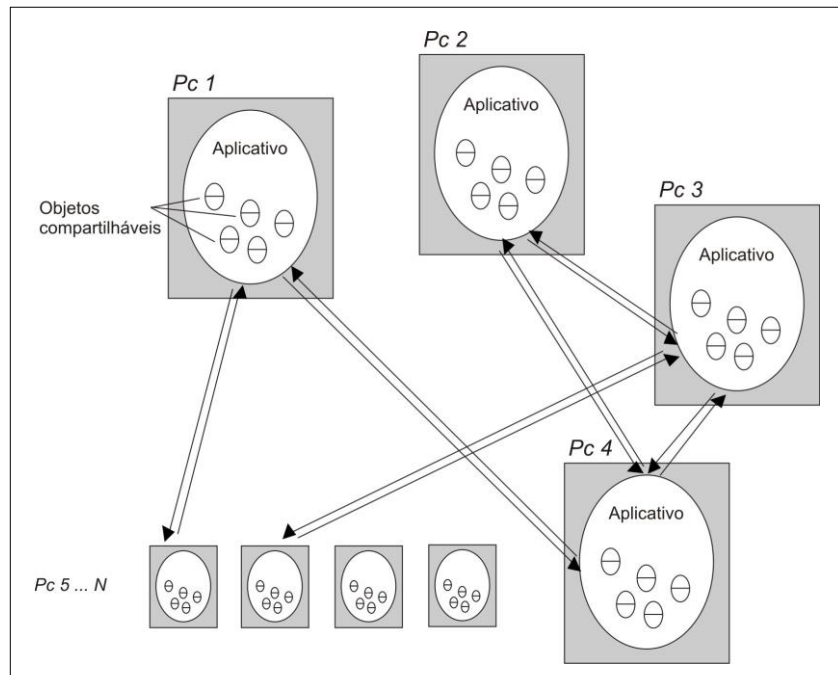
Na Figura 4 (B) é apresentado o desmembramento do arquivo 'a' em três porções, armazenadas em servidores diferentes: servidor 01, 02 e 03. O mesmo acontece com o arquivo 'e', sendo armazenadas as porções nos servidores 01, 02 e 03 (TANENBAUM; STEEN, 2007).

2.3.3 Arquitetura *peer-to-peer*

O modelo de arquitetura *peer-to-peer*, apresentado na Figura 5, explora recursos de *hardware* ou *software* dos computadores pertencentes ao sistema de arquivos distribuído para que possa realizar uma tarefa (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Segundo Coulouris, Dollimore e Kindberg (2007) nesta arquitetura todo computador opera simultaneamente como cliente e servidor, não havendo centralização. Esta arquitetura visa, entre outras vantagens, reduzir o custo de projeto e fazer com que as mensagens sejam trocadas entre os nós, em vez de priorizar o tráfego de mensagens entre um servidor centralizado e seus clientes.

Figura 5 – Aplicativo baseado em arquitetura *peer-to-peer*



Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2007)

2.4 Considerações finais

Nesta seção foram apresentadas a conceituação e a motivação para utilização de sistemas de arquivos distribuídos, seguidas da apresentação das principais características que são levadas em conta quando da concepção de um sistema deste tipo.

Dentre os itens apresentados e que merecem destaque estão a disponibilidade, a tolerância à falhas e a replicação. Itens estes utilizados para propor o melhoramento na questão da disponibilidade do sistema FlexA original, foco deste trabalho.

Também foram apresentadas as possíveis arquiteturas que um sistema de arquivos distribuído pode adotar sendo a arquitetura cliente-servidor, a arquitetura baseada em *cluster* e a arquitetura *peer-to-peer*.

Os itens abordados neste Capítulo serviram de base para o estudo e a comparação dos sistemas de arquivos distribuídos apresentados no Capítulo 3.

3 ESTUDOS DE CASOS

3.1 Considerações iniciais

Neste capítulo são apresentados estudos de casos de sistemas de arquivos distribuídos com base nos itens abordados na seção anterior.

Inicialmente, é apresentado o sistema de arquivos distribuído NFS – *Network File System* (sistema concebido em meados de 1980), seguido do sistema AFS – *Andrew File System* (que utiliza a arquitetura cliente-servidor). O GFS – *Google File System* foi concebido pela Google com base na arquitetura em *cluster*, sendo apresentado na seção 3.4. A seguir, são apresentados os sistemas Lustre e Tahoe-LAFS. Por fim, é abordado o sistema FlexA original; sistema que tem como características flexibilidade, desempenho e utilização de *hardware* de baixo custo.

3.2 NFS - *Network File System*

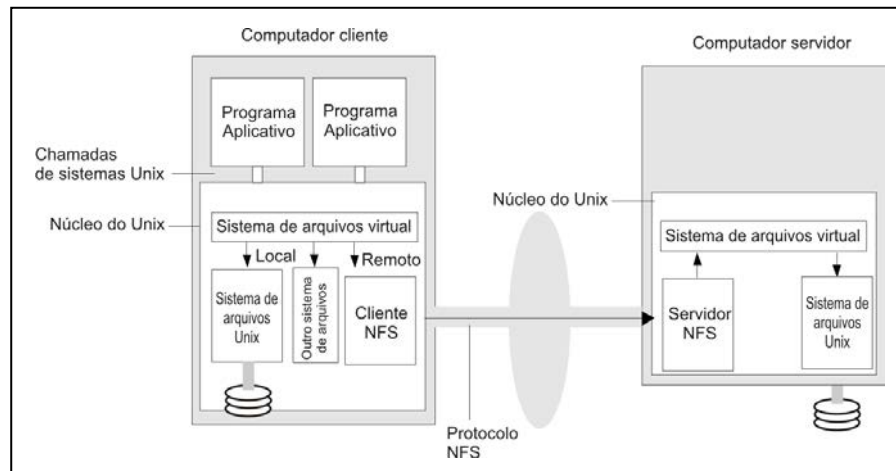
O *Network File System* da *Sun Microsystems* foi introduzido em 1985. Visando a adoção do NFS como padrão, suas interfaces foram disponibilizadas em domínio público, possibilitando o desenvolvimento de novas funcionalidades por outros fornecedores (COULOURIS; DOLLIMORE; KINDBERG, 2007).

3.2.1 Arquitetura

O *Network File System* é composto por três itens essenciais: protocolo de comunicação, módulo servidor e módulo cliente (OSADZINKI, 1998).

Na Figura 6 é apresentada a arquitetura do *Network File System*.

Figura 6 – Arquitetura do NFS



Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2007)

O NFS usa como protocolo de comunicação o mecanismo RPC (*Remote Procedure Call*), desenvolvido para suportar serviços distribuídos. Este mecanismo pode utilizar os protocolos TCP (*Transmission Control Protocol*) ou UDP (*User Datagram Protocol*) (TANENBAUM; STEEN, 2007; MULLER; MARLET, 1997).

O módulo servidor NFS está disponível no núcleo dos computadores que atuam como servidores, oferecendo uma interface capaz de receber requisições dos clientes (SHEPLER et al., 2003; COULOURIS; DOLLIMORE; KINDBERG, 2007).

O servidor NFS não mantém arquivos abertos em nome de clientes. A cada acesso é necessário verificar o endereço IP (*Internet Protocol*) do usuário e sua permissão para que o conteúdo possa ser acessado. Para evitar que requisições sejam manipuladas e enviadas em nome de usuários não autorizados, as requisições RPC são enviadas de forma criptografada (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Por intermédio dos clientes, as operações de leitura e escrita podem ser realizadas. Dessa forma, a transparência de acesso é feita através do módulo denominado Sistema de Arquivo Virtual (*VFS – Virtual File System*), possibilitando que clientes executem operações locais ou remotas sem distinção (OSADZINKI, 1998).

3.2.2 Cache e segurança

Este sistema de arquivos distribuído utiliza *cache* no servidor e no cliente. No servidor, os arquivos lidos recentemente são mantidos em *cache* na memória principal. Dessa

forma, em uma requisição, o conteúdo da *cache* é consultado antes de fazer um novo acesso ao disco.

Nos clientes, os resultados das operações são registrados em *cache*, visando reduzir o número de requisições ao servidor. Para garantir a utilização de um arquivo recente, o cliente consulta o servidor para verificar se a versão do arquivo que ele armazena é a mais atual. Essa técnica de validar o bloco de arquivos antes de utilizá-lo é chamada de *timestamp* (COULOURIS; DOLLIMORE; KINDBERG, 2007).

3.3 AFS - Andrew File System

O *Andrew File System* foi desenvolvido através de uma parceria entre a IBM (*International Business Machines*) e a CMU (*Carnegie Mellon University*), tendo a característica de trabalhar com centenas de estações vinculadas ao sistema. Este sistema agrega características dos sistemas de arquivos distribuídos, tais como transparência, mobilidade e segurança (HOWARD, 1988).

Segundo Tanenbaum e Steen (2007), o AFS visa a um bom desempenho quando comparado a outros sistemas de arquivos distribuídos que trabalham com um número maior de usuários e estações.

A seguir, é descrita a arquitetura do *Andrew File System* bem como outros itens relevantes associados a este sistema de arquivos distribuído.

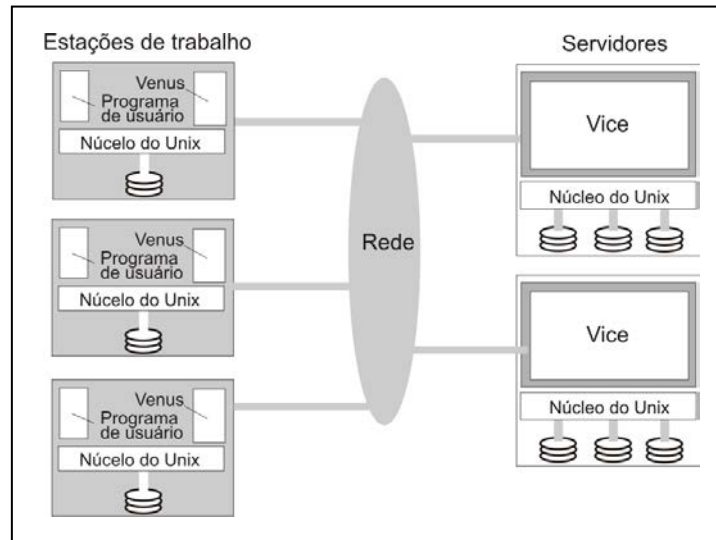
3.3.1 Arquitetura

O *Andrew File System* tem como base dois componentes de *software* que existem como processos *Unix*, denominados Vice e Vênus (COULOURIS; DOLLIMORE; KINDBERG, 2007).

Na Figura 7 é apresentada distribuição dos processos Vice e Venus. O Vice se refere ao *software* servidor executado no servidor. O processo Venus é executado no cliente (COULOURIS; DOLLIMORE; KINDBERG, 2007).

O AFS não oferece a possibilidade de compartilhamento de dados que esteja no sistema padrão *Unix*; ao invés disso, o servidor armazena os dados em formato especial para que possa ser acessado pelo uso de pedidos expedidos pelos clientes (TOBBICKE, 1994).

Figura 7 – Distribuição de processos no AFS



Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2007)

Neste sistema de arquivos distribuído, os arquivos disponíveis nas estações podem ser de dois tipos: locais ou compartilhados. Os arquivos locais são tratados como arquivos normais do *Unix*; já os compartilhados ficam armazenados em servidores, sendo feito um mapeamento para os clientes na hierarquia de diretório *Unix*.

Cada arquivo ou diretório contido no espaço compartilhado é identificado por um identificador de arquivo (*FID - File Identifier*). Quando uma requisição é enviada do cliente para o servidor, este *FID* deve ser mantido para que o processo *Venus* possa identificar a qual arquivo se refere a solicitação. Os arquivos são agrupados em volumes tendo-se como base o seu tipo. No cliente, quando existe uma chamada do sistema a um arquivo compartilhado, esta chamada é interceptada e enviada ao processo *Venus*, para que possa ser tratada. Caso o arquivo não esteja disponível em *cache* local, é enviada uma mensagem ao servidor solicitando uma cópia do arquivo.

Ao final do processo, se o arquivo foi modificado, uma cópia é devolvida ao servidor e o arquivo original é mantido no cliente para futuras modificações.

3.3.2 *Cache* e segurança

Este sistema de arquivos distribuído utiliza *cache* no cliente. Dessa forma, o cliente utiliza espaço em disco para que possa armazenar cópias dos arquivos compartilhados, sendo responsável por remover arquivos menos utilizados da *cache* e alocar novos arquivos.

A consistência da *cache* é mantida por meio de mensagens que são enviadas entre o servidor e o cliente. Nesse caso, quando o Vice fornece uma cópia para o processo Vênus, também fornece a promessa de *callback* (retorno) garantindo assim, que o Vênus seja avisado quando outro cliente modificar um arquivo (COULOURIS; DOLLIMORE; KINDBERG, 2007; TANENBAUM; STEEN, 2007). A segurança faz uso de RPC seguras.

3.4 GFS - Google File System

O *Google* desenvolveu seu próprio sistema de arquivos distribuído baseado na arquitetura em *cluster*, o *GFS* (*Google File System*). A motivação para a criação de seu próprio sistema é que os arquivos utilizados pelo *Google* são demasiadamente maiores (chegando a alcançar a casa dos *gigabytes*); no entanto, na maioria das vezes os arquivos são modificados por anexação de conteúdo (TANENBAUM; STEEN, 2007).

A ideia de funcionamento do GFS é que um arquivo seja dividido em porções de tamanho fixo (identificadas a partir de um identificador) e armazenado em servidores de porção. Para que a disponibilidade possa ser aumentada, as porções dos arquivos são replicadas em outros servidores de porção (por padrão três porções, podendo ser configurado para mais estações) (TANENBAUM; STEEN, 2007).

Este sistema disponibiliza uma interface que suporta operações sobre os arquivos, estes identificados hierarquicamente em diretórios (GHEMAWAT; GOBIOFF; LEUNG, 2003).

3.4.1 Arquitetura

Um *cluster Google File System* é composto de três elementos essenciais: um servidor mestre (*GFS master*), múltiplos servidores de porção (*GFS chunkserver*) e múltiplos clientes (*GFS client*).

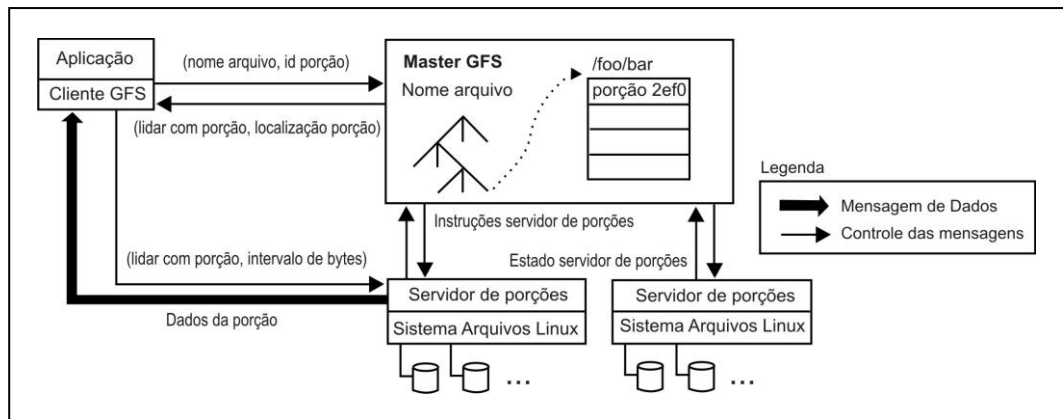
O servidor mestre armazena *metadados* dos arquivos armazenados, sendo: nome do arquivo, nome das porções dos arquivos e mapeamento dos arquivos para porção (a localização das porções de um arquivo). O servidor mestre também controla a migração de porções entre servidores de porção, faz o gerenciamento de porções órfãs e realiza a coleta de lixo (GHEMAWAT; GOBIOFF; LEUNG, 2003).

O papel dos clientes nessa estrutura é a comunicação com o servidor mestre e com os servidores de porção nas operações de leitura e escrita de arquivos. Os clientes nunca fazem

operações de leitura ou escrita diretamente no servidor de porções; ao invés disso, é solicitado ao servidor mestre a localização das porções e seus respectivos servidores de armazenamento.

Cabe aos servidores de porção armazenar porções do arquivo após seu desmembramento. Na Figura 8 é apresentada a arquitetura do GFS.

Figura 8 – Arquitetura do *Google File System*



Fonte: Adaptado de Ghemawat, Gobioff e Leung (2003)

Na Figura 8 são representados um cliente, um servidor *master* e dois servidores de porção. No ato da leitura de um arquivo, o cliente solicita os *metadados* ao servidor *master* (que detém as informações de localização do arquivo), sendo enviadas ao cliente. De posse dos *metadados* do arquivo, a comunicação é realizada junto aos servidores de porção para a solicitação das mesmas.

No ato da escrita, cabe ao *master* fazer a divisão do arquivo em porções de tamanho fixo; porções estas identificadas por um identificador global. Após a divisão do arquivo, as porções são enviadas aos servidores de porção, que fazem a replicação para outros três servidores de porção.

Nessa estrutura, o cliente não possui arquivos em *cache*, o que fica armazenado em *cache* são somente os *metadados* dos arquivos utilizados por um tempo determinado para que possam interagir diretamente com os servidores de porção, ao invés de consultar o servidor *master* novamente para obtenção da localização dos arquivos (GHEMAWAT; GOBIOFF; LEUNG, 2003; OSADZINSKI, 2010).

3.4.2 Disponibilidade

O GFS mantém o sistema disponível através das técnicas de recuperação rápida e de replicação de dados.

A técnica de recuperação rápida faz com que servidores *master* ou servidores de porções executem processos de recuperação após falhas para que, em segundos, tais servidores estejam disponíveis para voltar a oferecer serviços (COULOURIS; DOLLIMORE; KINDBERG, 2007).

A técnica de replicação de dados está relacionada à replicação de porções e replicação do servidor *master*. No primeiro caso, as porções são replicadas para três servidores de porção, podendo ser modificado o número de porções na configuração do sistema. No caso da replicação do servidor *master*, é feita uma cópia de seu estado nos servidores de porção. No caso de falha do servidor *master*, os servidores de porção são contatados (GHEMAWAT; GOBIOFF; LEUNG, 2003).

3.5 Lustre

O sistema de arquivos distribuído Lustre é baseado na plataforma *open source* Linux (JIAN; ZHAN-HUAI; XIAO, 2012). Este sistema é utilizado por diversos tipos de *cluster*, tendo destaque devido à capacidade de trabalhar com dezenas ou centenas de clientes e pela capacidade de trabalhar com um conteúdo grande de dados (ORACLE Corporation, 2014). Uma característica deste sistema é que os *metadados* são desacoplados das operações de entrada e saída de um arquivo, uma vez que os *metadados* e os dados de um arquivo são armazenados em servidores distintos (LOGAN; DICKENS, 2008).

3.5.1 Arquitetura

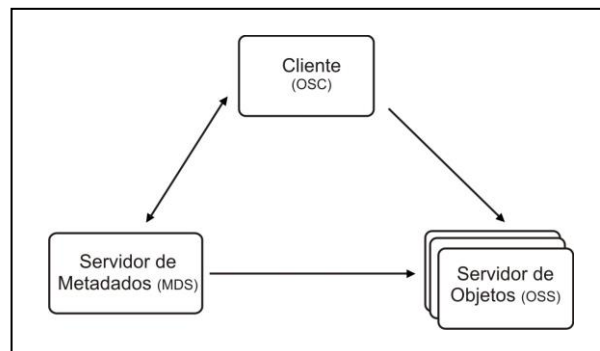
A arquitetura do Lustre consiste em três componentes: os clientes do sistema de arquivos (OSC - *Object Storage Client*), os servidores de armazenamento de objetos - também conhecidos como arquivos (OSS - *Object Storage Servers*) e os servidores de *metadados* (MDS - *Metadata Services*).

A função dos clientes é a de requisitar serviços do tipo escrita e leitura de arquivos.

Os servidores de arquivos têm a função de prover o armazenamento de arquivos, atendendo às operações de escrita e leitura de arquivos (JIAN; ZHAN-HUAI; XIAO, 2012).

Por fim, o servidor de *metadados* (MDS - *Metadata Server*) armazena as informações dos arquivos, gera o espaço de nomes e identifica o endereço de armazenamento dos objetos (YU et al., 2006). Na Figura 9 é apresentada a arquitetura do Lustre.

Figura 9 – Arquitetura do Lustre



Fonte: Adaptado de Jian, Zhan-Huai e Xiao (2012)

Sob a ótica do cliente, Yu et al. (2006) evidenciam que, para acessar um arquivo, inicialmente o cliente deve obter do servidor de *metadados* as informações do arquivo. Após a obtenção dessas informações, as operações de entrada e saída de arquivos são realizadas diretamente entre o cliente e o servidor de objetos.

A disponibilidade do Lustre está ligada à recuperação do cliente e do servidor de *metadados*. A recuperação de uma falha do cliente é baseada em bloquear a revogação de recursos para clientes que falharam, fazendo com que clientes sobreviventes possam continuar os trabalhos de uma forma ininterrupta.

No caso de falha do servidor de *metadados* (reconhecida por verificações que são realizadas durante a execução do sistema), um novo servidor de *metadados* de *backup* é eleito com base nos servidores de arquivos disponíveis para continuar a oferecer o serviço pertinente ao servidor de *metadados* (ORACLE Corporation, 2014).

3.6 Tahoe-LAFS

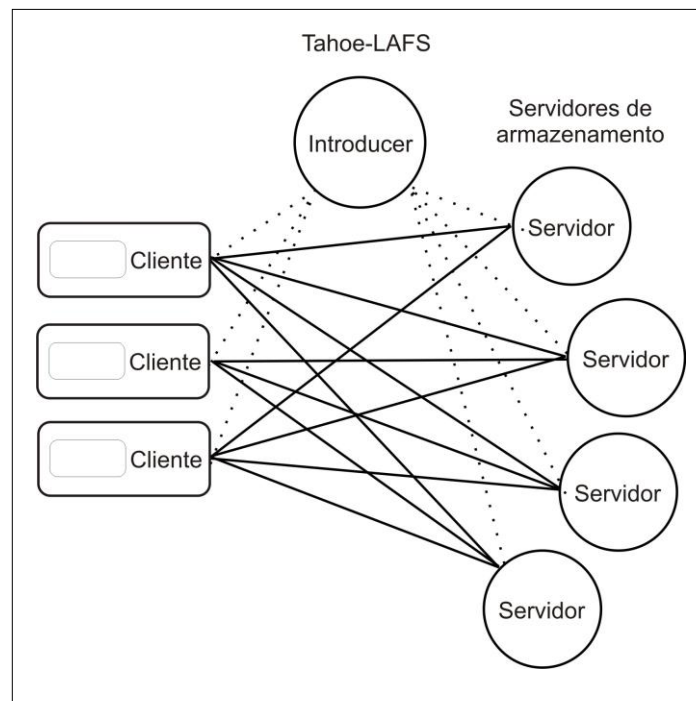
De código aberto, desenvolvido pela *allmydata.com* e implementado em *Python*, este sistema foi lançado em 2006. Atualmente, é distribuído através do sítio do Tahoe-LAFS (TAHOE-LAFS, 2014).

O Tahoe-LAFS oferece como características de um sistema de arquivos distribuído: confidencialidade, integridade e disponibilidade. Além disso, faz uso do princípio de menor autoridade, ao qual o usuário realiza as tarefas sem obter mais autoridade do que é necessário (WILCOX-O'HEARN; WARNER, 2008).

3.6.1 Arquitetura

A arquitetura deste sistema de arquivos distribuído tem três elementos essenciais: um componente central denominado *Introducer*, servidores de armazenamento e clientes. Na Figura 10 é apresentada a interação entre os elementos essenciais do Tahoe-LAFS.

Figura 10 – Interação entre os elementos essenciais do Tahoe-LAFS

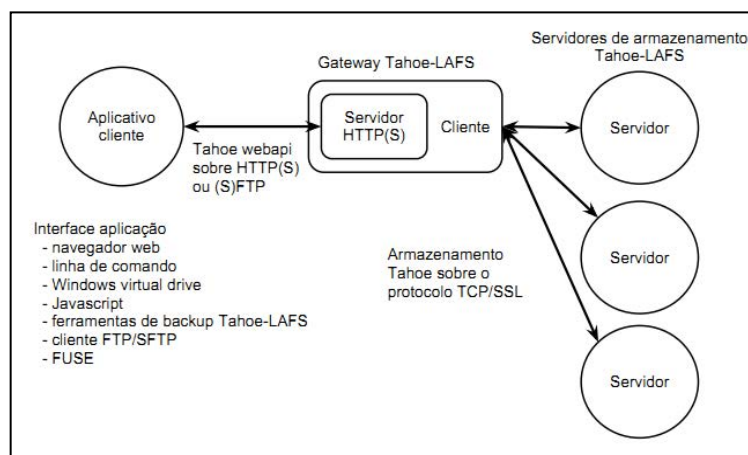


Fonte: Adaptado de TAHOE-LAFS (2014)

A função do *Introducer* é registrar a localização de todos os nós que fazem parte do sistema; no entanto, a transferência dos dados é realizada diretamente entre clientes e servidores de armazenamento.

Os servidores de armazenamento têm a função de armazenar porções de arquivos na operação de escrita e oferecer tais porções na operação de leitura. Cabe aos clientes realizar as operações de escrita e leitura. Na Figura 11 é apresentada uma visão geral do Tahoe-LAFS.

Figura 11 – Visão geral do Tahoe-LAFS



Fonte: Adaptado de TAHOE-LAFS (2014)

Cada cliente Tahoe-LAFS tem a presença de um *gateway* que garante a criptografia e a integridade dos arquivos que são trocados entre cliente e servidor. A troca de informação pode ser feita através dos protocolos HTTP (*HyperText Transfer Protocol*) ou FTP (*File Transfer Protocol*), navegador *web* ou *prompt* de comandos.

Depois da criptografia dos arquivos pelo *gateway*, os dados são distribuídos para os servidores de armazenamento, respeitando o ajuste no processo de distribuição $1 \leq K \leq N$, em que N representa a quantidade de servidores que serão utilizados para armazenar os arquivos e K o mínimo de servidores necessários para disponibilizar o arquivo.

3.6.2 Disponibilidade

O Tahoe-LAFS garante a disponibilidade através da maneira como os arquivos são armazenados. Por padrão são necessários 10 servidores para armazenar os arquivos na operação de escrita e no mínimo 3 servidores para resgatá-los na operação de leitura. No entanto, o número de servidores para armazenamento pode ser modificado no caso de indisponibilidade desta quantidade de servidores (WILCOX-O'HEARN; WARNER, 2008).

Segundo Wilcox-o e Warner (2008 citado por Fernandes, et al. 2012), na operação de escrita, o Tahoe-LAFS realiza a criptografia do arquivo, divide-o em porções (dependendo da configuração do sistema) e envia essas porções para os servidores disponíveis. Na leitura de um arquivo, mesmo que um ou mais servidores estejam indisponíveis, outro servidor será responsável por oferecer a porção do arquivo necessária para que o arquivo possa ser composto.

3.7 FlexA original

O sistema de arquivos distribuído **Flexível e Adaptável** – FlexA original foi concebido em *Python* com base em características herdadas dos sistemas de arquivos distribuídos, Tahoe-LAFS, GFS (*Google File System*), AFS (*Andrew File System*) e NFS (*Network File System*).

O conceito de permissões de arquivo e uso de *hardware* de baixo custo foi uma característica herdada do Tahoe-LAFS devido ao fato do FlexA original ter sido desenvolvido em *Python*; a disponibilidade está ligada a questão da replicação de porções de arquivos, característica esta herdada do *Google File System* e do Tahoe-LAFS; a utilização de *cache*, foi herdada do AFS e, por fim, tornar o sistema de fácil utilização, foi um conceito abstraído do NFS.

Dentre as características do FlexA original, destacam-se: transparência, tolerância a falhas, criptografia, capacidade de trabalhar com *hardware* de baixo custo e facilidade na manipulação de arquivos (FERNANDES, 2012).

A seguir, são apresentadas a arquitetura do FlexA original e suas principais características.

3.7.1 Arquitetura

A arquitetura do FlexA original é diferente do modelo cliente-servidor, uma vez que não existe a presença de um servidor principal na sua arquitetura. Ao invés disso, as estações presentes nos grupos que fazem parte do sistema comunicam-se para a descoberta de novas estações de trabalho, aproximando-se do padrão *peer-to-peer*.

No FlexA original, as estações de trabalho pertencentes ao sistema de arquivos distribuído podem estar inseridas em um dos três grupos que compõem o sistema: Grupo de Escrita (que contém servidores primários), Grupo de Réplicas (que contém servidores secundários) ou Grupo de Clientes (que contém clientes). Na Figura 12 é apresentada a arquitetura do FlexA original.

O Grupo de Escrita é responsável por administrar e armazenar arquivos com suas informações (*metadados*). Este grupo é composto por, no mínimo, três computadores para que possam ser garantidas a disponibilidade e a distribuição dos arquivos.

No Grupo de Réplicas os computadores podem desempenhar somente a função de servidores secundários ou de servidores secundários e servidores primários ao mesmo tempo.

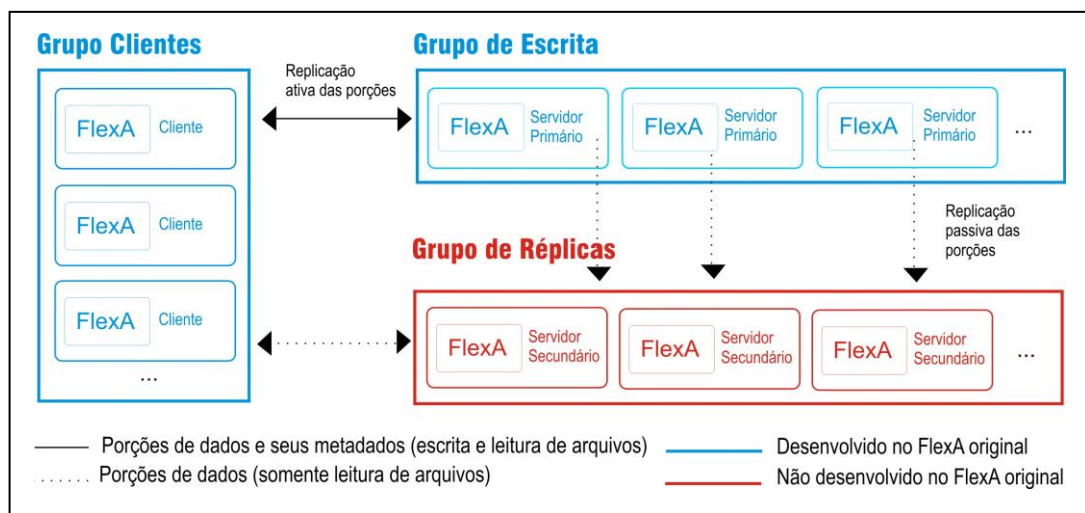
No primeiro caso, ao desempenhar a função de servidor secundário, este atuará como servidor de *backup* de porções enviadas a partir dos servidores primários. No segundo caso, além de oferecer o armazenamento de porções de *backup*, poderá atuar como servidor primário, caso seja eleito na falha de um servidor primário.

O Grupo de Réplicas é formado por, no mínimo, dois servidores secundários devido o fato de que, ao receber uma porção, o servidor primário deve escolher dois servidores secundários para enviar cópias da porção. Recomenda-se um número maior de servidores, caso seja possível, para que possam atuar como servidores primários na sobrecarga de servidores primários.

O Grupo de Réplicas está representado na Figura 12, no entanto não foi desenvolvido na versão do FlexA original, sendo um dos itens ligados à proposta deste trabalho e que será discutido posteriormente, no capítulo 4.

O Grupo de Clientes possui estações de trabalho do tipo cliente, que fazem a interação entre o sistema de arquivos distribuído e o usuário, sendo responsável por realizar operações sobre os arquivos, tais como escrita, leitura, modificação de permissão e listagem de arquivos.

Figura 12 – Arquitetura do FlexA original



Fonte: Adaptado de Fernandes (2012)

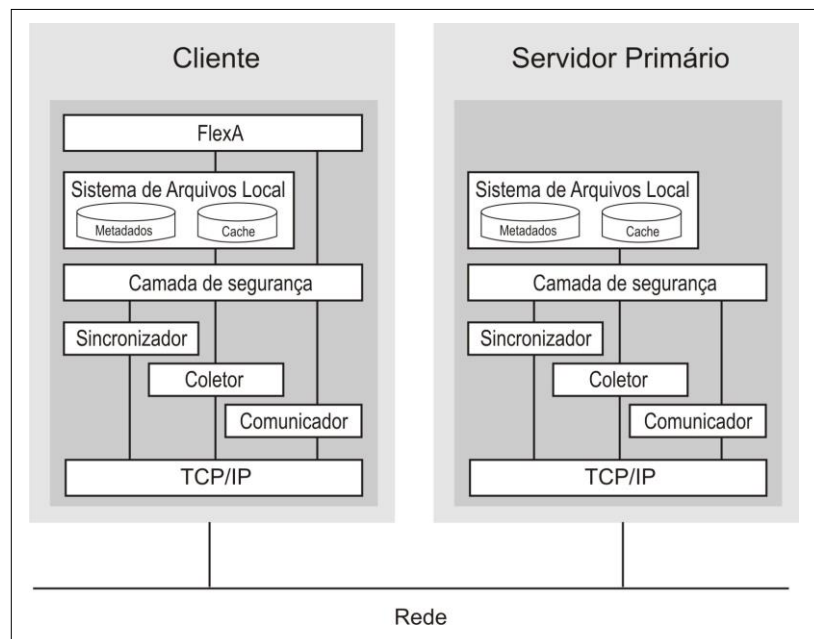
Como exemplo de funcionamento do sistema, na operação de escrita no FlexA original, a função do cliente é fazer a criptografia dos arquivos, dividir o arquivo em porções

(por padrão três porções) e distribuir essas porções para os servidores primários. Na operação de leitura, o cliente verifica as permissões do arquivo, solicita porções que fazem parte do arquivo aos respectivos servidores primários, junta as porções e decifra o arquivo, disponibilizando-o ao usuário. No grupo de clientes, cada cliente é responsável pela permissão de seus arquivos, não tendo a intervenção de administradores ou usuários privilegiados.

3.7.2 Módulos

O FlexA original contém três módulos: Coletor, Sincronizador e Comunicador. Esses módulos compõem o núcleo do sistema, sendo que a comunicação entre os grupos é feita a partir de *sockets* sobre o protocolo TCP/IP (*Transmission Control Protocol/Internet Protocol*). Na Figura 13 são apresentados os módulos do FlexA original.

Figura 13 – Módulos do FlexA original



Fonte: Adaptado de FERNANDES (2012)

Módulo Coletor - O módulo Coletor é responsável pela interação entre clientes, servidores primários e servidores secundários. Cabe ao módulo Coletor receber todas as requisições de entrada do sistema e fazer os devidos encaminhamentos.

Módulo Sincronizador - O módulo `Sincronizador` é utilizado sob demanda. A função deste módulo é a de informar através de mensagens, clientes, servidores primários e servidores secundários de que os dados armazenados sofreram modificação.

Módulo Comunicador - Pelo fato do FlexA original não ter o papel de um servidor central, faz-se necessário um módulo denominado `Comunicador`.

Este módulo é carregado quando o sistema é iniciado, fazendo a busca na rede pelos computadores que estão com o módulo `Coletor` ativo e que pertençam a alguns dos grupos (Escrita, Réplicas ou Clientes) do FlexA original. Ao localizar uma estação, são registrados o endereço IP, UUID (*Universally Unique Identifier*) e o tipo de estação (cliente, servidor primário ou servidor secundário). A busca por novas estações acontece a cada trinta segundos, após o início deste módulo.

3.7.3 Flexibilidade

A flexibilidade do FlexA original está ligada a possibilidade de modificação de itens essenciais que compõem o sistema pelo fato deste sistema ter sido desenvolvido em linguagem de programação *open source*, *Python*.

Um exemplo é a possibilidade de alteração do algoritmo de criptografia, a alteração dos níveis de replicação ou até mesmo a definição de uma nova interface para o sistema.

Pelo fato do sistema agregar esta característica, é que um novo conceito de como lidar com a questão da disponibilidade pôde ser apresentado neste trabalho.

3.7.4 Utilização de *hardware* de baixo custo

Em um sistema de arquivos distribuído que utiliza a arquitetura cliente-servidor, nota-se a centralização das operações no servidor.

No FlexA original, essa ordem se inverte, de modo que o cliente passa a ser responsável pela maioria das operações que tornam um arquivo disponível (criptografia, divisão do arquivo em porções e envio aos servidores).

Neste sistema, a carga de trabalho fica distribuída entre os clientes, fazendo com que os servidores tenham uma carga de trabalho menor e, conseqüentemente, um consumo de *hardware* menor (FERNANDES et al., 2013).

3.7.5 Desempenho

No intuito de melhorar o desempenho na transferência de arquivos, o FlexA original faz uso de *cache* no cliente, para que seja evitado o envio de um arquivo ao servidor, caso esteja armazenado em *cache* (FERNANDES, 2012).

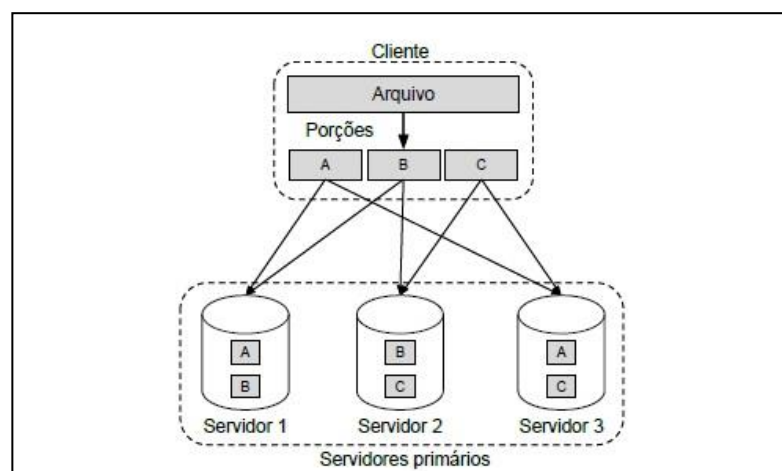
Dessa forma, o FlexA original reflete a mesma forma de armazenamento em *cache* no cliente do *Andrew File System*, descrita por Coulouris, Dollimore e Kindberg (2007).

3.7.6 Controle de acesso e disponibilidade

O controle de acesso do FlexA original foi concebido tendo-se como base o modelo de permissões do sistema de arquivos distribuído Tahoe-LAFS, que trata arquivos e diretórios de forma individual. As permissões de arquivos e diretórios são manipuladas diretamente pelos usuários sem a necessidade de obtenção de usuário e senha; ao invés disso, cada cliente possui dois manipuladores (somente leitura ou leitura e escrita) com chave de criptografia e validação para a administração dos arquivos.

A garantia de disponibilidade no FlexA original está associada a dois itens: a utilização *cache* no cliente e a forma com que as porções são enviadas e mantidas nos servidores. A Figura 14 apresenta a forma na qual um arquivo é escrito e lido no FlexA original.

Figura 14 – Leitura e escrita no FlexA original



Fonte: Adaptado de FERNANDES (2012)

Na Figura 14, o cliente realiza a criptografia e divisão do arquivo de nome ‘Arquivo’ em três porções (A, B e C), enviando 2/3 do arquivo para cada um dos servidores primários disponíveis. Cabe aos servidores primários armazenarem as porções dos arquivos através da operação de escrita e disponibilizar tais arquivos através da operação de leitura, mesmo no caso em que um dos servidores primários não estiver ativo. No caso de falha de mais de um servidor primário, o cliente é informado da indisponibilidade do serviço (FERNANDES, 2012; FERNANDES et al., 2013). Isto se dá pelo fato de que, com somente um servidor ativo, não é possível ter as três porções necessárias para compor o arquivo.

3.8 Considerações finais

Neste capítulo foram apresentados estudos de casos dos sistemas de arquivos distribuídos NFS, AFS, GFS, Lustre, Tahoe-LAFS e do FlexA original.

Os sistemas de arquivos distribuídos NFS e AFS foram concebidos baseados na arquitetura cliente-servidor, fazendo uso de *cache* e realizando comunicação através de RPCs seguras. A arquitetura baseada em *cluster* foi utilizada na concepção dos sistemas de arquivos distribuídos GFS, Tahoe-LAFS e FlexA original, sendo realizada a divisão do arquivo em porções e envio a servidores de armazenamento. Por fim, o sistema Lustre utiliza arquitetura híbrida baseada nas arquiteturas cliente-servidor e *cluster*, sendo que os clientes necessitam da obtenção de privilégios junto ao servidor de *metadados* para realizar as operações sobre o arquivo.

No próximo capítulo são apresentados os melhoramentos no que diz respeito à característica de disponibilidade do sistema de arquivos distribuído FlexA original.

4 TRABALHO DESENVOLVIDO

No desenvolvimento do FlexA original, a garantia da disponibilidade do sistema está ligada à manutenção de *cache* local no cliente e na forma como um arquivo é armazenado. Com relação à forma como o arquivo é armazenado, na operação de escrita, o arquivo é criptografado e dividido em 3 porções pelo cliente, sendo enviadas porções na ordem de 2/3 para cada servidor primário.

Neste cenário, na ausência de um servidor primário, o sistema continua a operar somente na operação de leitura de arquivos. No entanto, na falha de mais de um servidor primário (o sistema atuando com somente um servidor primário), o cliente não tem a garantia de leitura ou escrita de um arquivo, fazendo com que a solução de disponibilidade seja limitada.

Além disso, no FlexA original não foram implementados meios de recuperação de servidores primários e tampouco realizados tratamentos na questão de sobrecarga de servidores primários e servidores secundários; itens estes que fazem com que o sistema se torne indisponível após sua ocorrência.

Neste trabalho, é apresentado o melhoramento da disponibilidade no sistema FlexA original, através do desenvolvimento dos itens:

- Grupo de Réplicas - desenvolvimento do Grupo de Réplicas para que a garantia de disponibilidade de arquivos passe a contar com este grupo de armazenamento. As operações de escrita e leitura sofreram modificações para se adequar a este grupo. O balanceamento de carga foi utilizado na escrita, sendo descrito por SEGURA, 2013.
- Autoavaliação de servidores primários - desenvolvimento de um método para verificação e recuperação de servidores primários após a detecção de falha por queda;
- Sobrecarga de servidores - desenvolvimento de uma forma de lidar com a detecção de sobrecarga de servidores primários e de servidores secundários através da

verificação de informações destes servidores (CARVALHO; LOBATO; MANACERO JUNIOR, 2013).

No Quadro 1 é apresentado um comparativo entre as funcionalidades presentes no FlexA original e as funcionalidades propostas para o FlexA desenvolvido.

Quadro 1 – Comparação entre o FlexA original e o FlexA desenvolvido

Característica	FlexA original	FlexA desenvolvido
Arquitetura	Grupo de Clientes e Grupo de Escrita	Grupo de Clientes, Grupo de Escrita e Grupo de Réplicas. Modificação das operações de escrita e leitura.
Disponibilidade	Divisão do arquivo em 3 porções pelo cliente e envio de 2/3 do arquivo para cada servidor primário	Divisão do arquivo em 3 porções pelo cliente e envio de uma porção do arquivo para cada um dos servidores primários ativos. Replicação usando o Grupo de Réplicas
Balanceamento de carga na operação de escrita	Não	Sim
Deteção e tratamento na indisponibilidade de servidores primários	Não	Autoavaliação dos servidores primários
Deteção de sobrecarga do sistema	Não	Avaliação de sobrecarga do servidor primário e servidor secundário

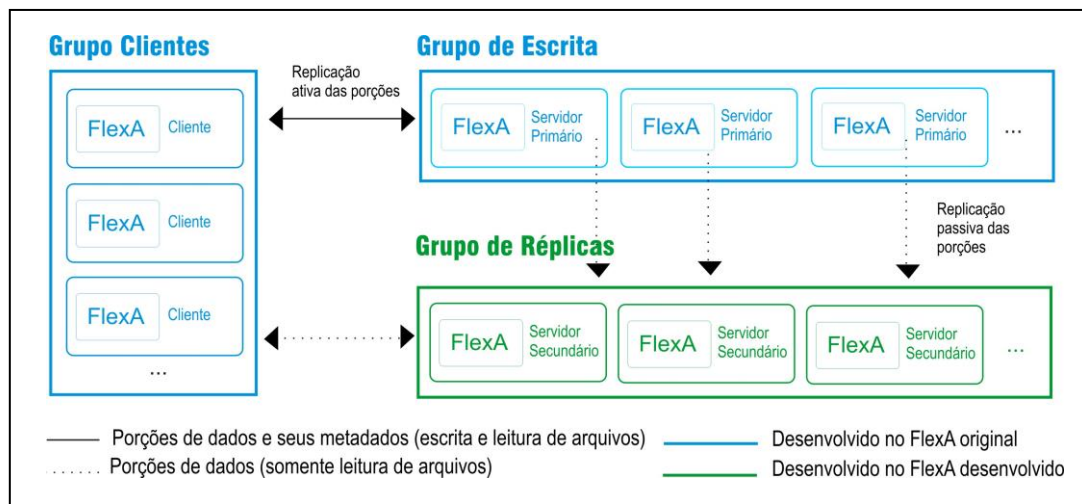
A seguir é apresentada arquitetura do FlexA desenvolvido após o acréscimo dos itens visando a melhora da sua disponibilidade, seguido da apresentação dos módulos e detalhes de como os itens ‘Grupo de Réplicas’, ‘autoavaliação de servidores primários’, ‘sobrecarga dos servidores primários’ e ‘sobrecarga dos servidores secundários’ foram incorporados no sistema FlexA original, resultando no FlexA desenvolvido.

4.1 Arquitetura

No sentido de propor uma forma de lidar com a questão da disponibilidade no FlexA original, a arquitetura do FlexA desenvolvido passou a contar efetivamente com o Grupo de Réplicas, uma vez que este grupo não foi implementado na versão do FlexA original. Dessa forma, a arquitetura do FlexA desenvolvido é composta pelos grupos de Clientes, Escrita e Réplicas.

Na Figura 15 é apresentada a arquitetura do FlexA desenvolvido, com a inclusão efetiva do Grupo de Réplicas.

Figura 15 – Arquitetura do FlexA desenvolvido



Fonte: SEGURA et al., 2013

Devido a agregação do Grupo de Réplicas no FlexA desenvolvido, os clientes deixaram de armazenar *metadados* de arquivos, função esta que passou a ser dos servidores primários.

Os servidores primários continuaram a desempenhar a mesma função mencionada no FlexA original: armazenando porções de arquivos. No entanto, passaram a armazenar *metadados* das porções. Além disso, tais servidores sofreram modificações para se adequar às operações de escrita e leitura, passando a replicar porções e a sincronizar *metadados*.

Os servidores secundários, pertencentes ao Grupo de Réplicas, passaram a ser responsáveis por receber *metadados* e porções dos servidores primários, armazenando-os. Este grupo é composto por, no mínimo, dois servidores secundários, podendo ser estendido a um número maior.

Módulos

Com o desenvolvimento do Grupo de Réplicas, os servidores secundários agregaram os módulos `Coletor`, `Sincronizador` e `Comunicador` para que passassem a se comunicar com o Grupo de Escrita e o Grupo de Clientes, já existentes.

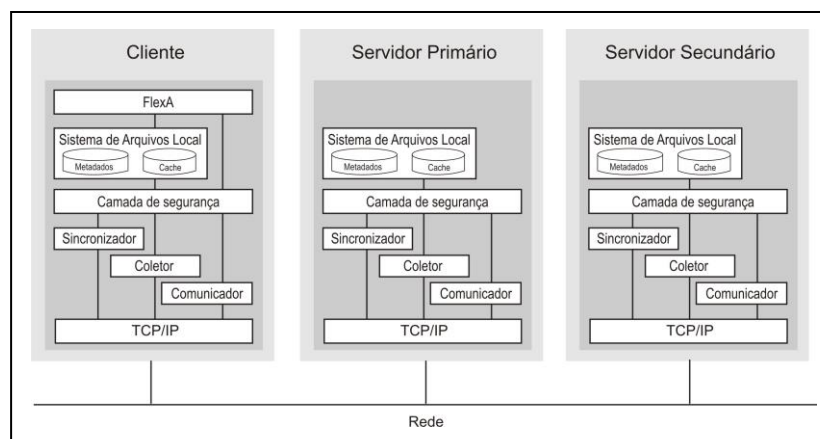
Dessa forma, o módulo `Coletor` continuou a desempenhar a mesma função mencionada no FlexA original, interagindo entre os grupos do sistema, recebendo todas as requisições de entrada e fazendo os devidos encaminhamentos. No entanto, sofreu modificações no sentido de agregar as novas funcionalidades de disponibilidade e tolerância a falhas, abordadas neste trabalho.

O módulo `Sincronizador` continuou a realizar a sincronização de servidores após a modificação de um arquivo; no entanto, foi adequado à nova arquitetura através de modificações.

O módulo `Comunicador` continuou a realizar a busca de novas estações e a mediar ações do sistema. No entanto, sofreu modificações no sentido de agregar as novas funcionalidades de disponibilidade e tolerância a falhas.

A Figura 16 apresenta os módulos do FlexA desenvolvido, após a inclusão do Grupo de Réplicas.

Figura 16 – Módulos do FlexA desenvolvido



Fonte: SEGURA et al., 2013

As modificações realizadas nos módulos `Coletor`, `Sincronizador` e `Comunicador` estão ligadas à forma de armazenamento de porções nos servidores primários (antes, 2/3 de um arquivo e neste modelo, uma porção para cada servidor),

modificações para verificar a sobrecarga de servidores primários e secundários e modificações para verificar falhas nos servidores primários.

As modificações realizadas para garantir uma melhor disponibilidade do sistema FlexA desenvolvido, são descritas com detalhes a seguir, na seção 4.2.

4.2 Operação de escrita

Após o desenvolvimento do Grupo de Réplicas, o FlexA desenvolvido passou a escrever os arquivos de forma diferente, se comparado ao FlexA original. Caracteriza-se como escrita, o processo de inserir (submeter, criar) um arquivo no sistema de arquivos distribuído.

Na operação de escrita do FlexA original, um arquivo é enviado a partir de qualquer cliente, que criptografa o arquivo usando uma chave de cifra do usuário, faz a divisão do mesmo em três porções iguais, e envia duas porções do arquivo para cada um dos três servidores primários ativos (deve-se ressaltar que, no caso de mais de três servidores primários disponíveis, somente três servidores serão escolhidos). No entanto, a disponibilidade deste sistema torna-se limitada pelo fato de que a queda de mais de um servidor primário ocasiona a indisponibilidade do sistema nas operações de leitura e escrita pelo fato de não haver mecanismos de recuperação de servidores primários. No caso da indisponibilidade de um servidor primário (e manutenção de dois servidores), o sistema passa a atuar somente na operação de leitura (FERNANDES, 2012).

O modelo de garantia de disponibilidade do sistema FlexA desenvolvido sofreu alterações, adotando o conceito de replicação de porções do arquivo através do uso do Grupo de Réplicas.

Dessa forma, na operação de escrita, um arquivo é submetido através de qualquer estação cliente que faça parte do sistema de arquivos distribuído. Nesta operação, inicialmente o cliente criptografa o arquivo usando uma chave de cifra do usuário, divide o arquivo em três porções e envia cada porção para três servidores primários distintos. É importante ressaltar que são necessários no mínimo três servidores primários e dois servidores secundários ativos no sistema para garantir a disponibilidade do sistema; na existência de mais de três servidores primários, três são escolhidos de forma aleatória. Na existência de mais de dois servidores secundários, dois são escolhidos com base em métricas destes servidores.

A operação de escrita continua com o recebimento das porções do arquivo pelos servidores primários e a replicação dessas porções para o grupo de réplicas, que contém os servidores secundários.

Antes de cada porção recebida por cada servidor primário ser replicada, é necessário escolher dois servidores secundários que receberão as porções do arquivo.

A escolha dos servidores secundários é feita com base em informações obtidas pelo sistema. Na prática, o servidor primário utiliza um algoritmo em anel para consultar informações dos servidores secundários e cria uma lista de candidatos utilizando a equação apresentada na Figura 17.

Figura 17 – Equação para coleta de métricas

$$\text{Métrica} = 1 - \left(\frac{\text{HD}}{3} + \frac{\text{DL}}{3} + \frac{\text{UL}}{3} \right)$$

$$\text{HD} = \frac{\text{Espaço em disco utilizado}}{\text{Capacidade do disco}}$$

$$\text{DL} = \frac{\text{Volume de tráfego de saída}}{\text{Capacidade do canal}}$$

$$\text{UL} = \frac{\text{Volume de tráfego de entrada}}{\text{Capacidade do canal}}$$

Fonte: Adaptado de SEGURA (2013)

O limite utilizado para selecionar candidatos é de 0,8 (ou 80% da ociosidade). Este limite é automaticamente atualizado se nenhum servidor secundário oferecer essa disponibilidade. Quando isso ocorre, o limite é reduzido em 0,1 (10%) a partir do valor anterior, até que um valor seja obtido. Seguindo essa política, o FlexA desenvolvido efetua o balanceamento de carga quando do envio das porções aos servidores secundários.

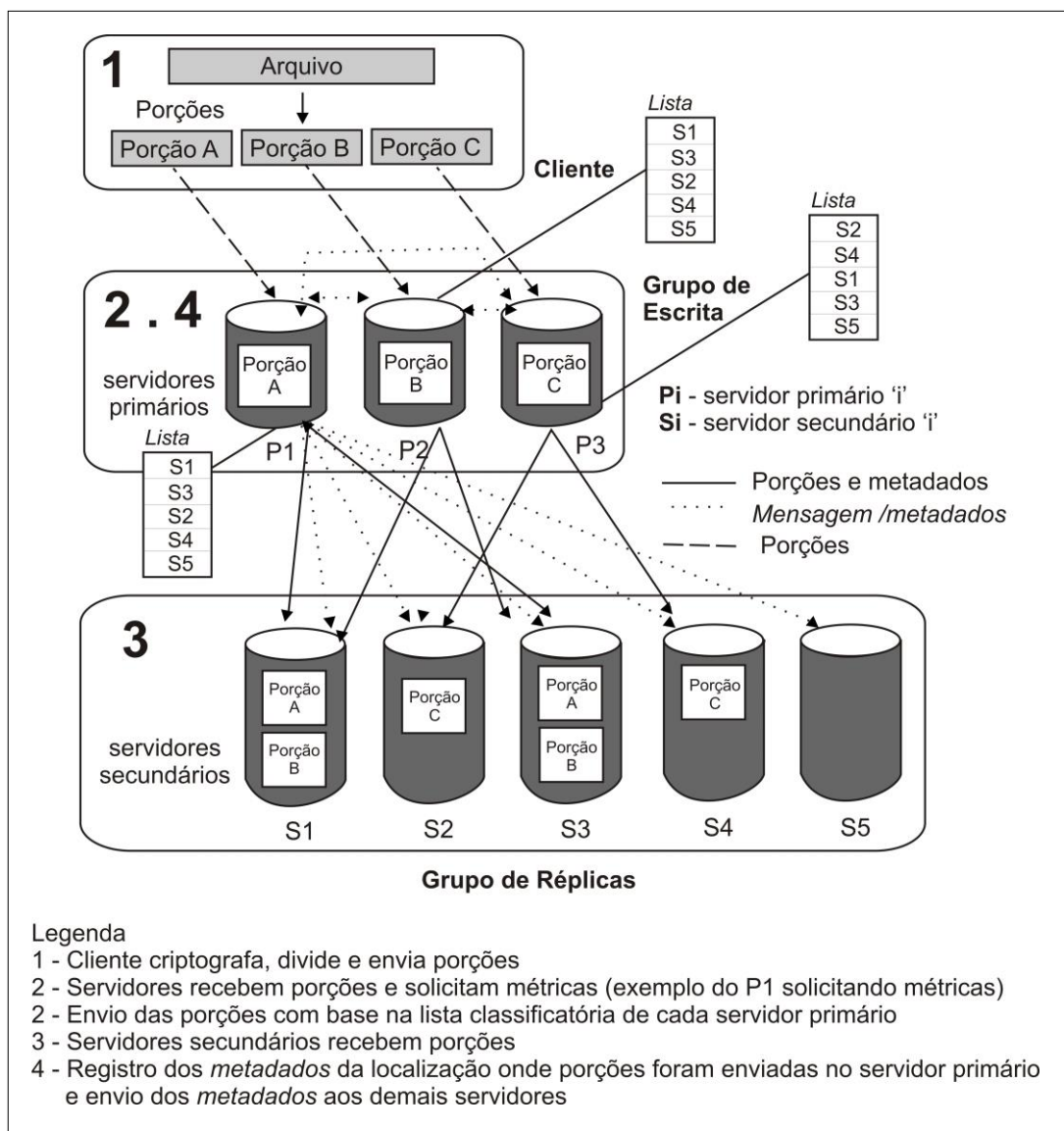
Os dois servidores secundários escolhidos são os primeiros da lista classificatória. Na Figura 18 é representada a operação de escrita no FlexA desenvolvido, na qual um arquivo de nome ‘Arquivo’ é criptografado pelo cliente e dividido em três porções, resultando nas porções ‘Porção A’, ‘Porção B’ e ‘Porção C’.

O servidor primário P1, após receber a ‘Porção A’, solicita as métricas de todos os servidores secundários e cria uma lista classificatória, visando o balanceamento no envio das porções aos servidores. As porções são enviadas aos dois primeiros servidores secundários da lista classificatória, neste exemplo, os servidores S1 e S3.

O servidor primário P2 envia a ‘Porção B’ para os servidores secundários S1 e S3 após realizar a coleta e a classificação dos servidores baseado nas métricas, e o servidor primário P3 envia a ‘Porção C’ para os servidores secundários S2 e S4, após realizar a coleta e a classificação dos servidores baseado nas métricas.

Após o envio das porções aos dois servidores secundários escolhidos, cada servidor primário registra os *metadados* dos servidores secundários para os quais as porções foram enviadas e, posteriormente, envia esses *metadados* aos servidores primários e aos servidores secundários. Mantendo os *metadados* atualizados em todos os servidores primários e secundários, todos os primários poderão oferecer *metadados* na solicitação dos *metadados* na operação de leitura; qualquer servidor secundário, uma vez eleito como servidor primário, também poderá oferecer *metadados* na operação de leitura.

Figura 18 – Operação de escrita no FlexA desenvolvido



4.3 Operação de leitura

No FlexA original, o cliente possuía os *metadados* dos arquivos, de tal forma que, no processo de leitura o cliente realizava uma requisição aos servidores de porção e, após o recebimento das porções que compõem o arquivo, as partes eram juntadas e descriptografadas.

Com as modificações no FlexA desenvolvido, na operação de leitura, um arquivo pode ser recuperado através de qualquer estação cliente que faça parte do sistema de arquivos distribuído. Nesta operação, inicialmente o cliente requisita os *metadados* do arquivo a um dos servidores primários ativos (como todos os servidores primários detêm os *metadados* dos arquivos, esta escolha é aleatória).

Antes de devolver os *metadados* para o cliente, o servidor primário verifica o espaço em disco e a ocupação do canal de comunicação de todos os servidores que detêm porções do arquivo (podendo ser servidores primários ou secundários), gerando uma lista de servidores ordenada com base na equação de métricas apresentada na Figura 17.

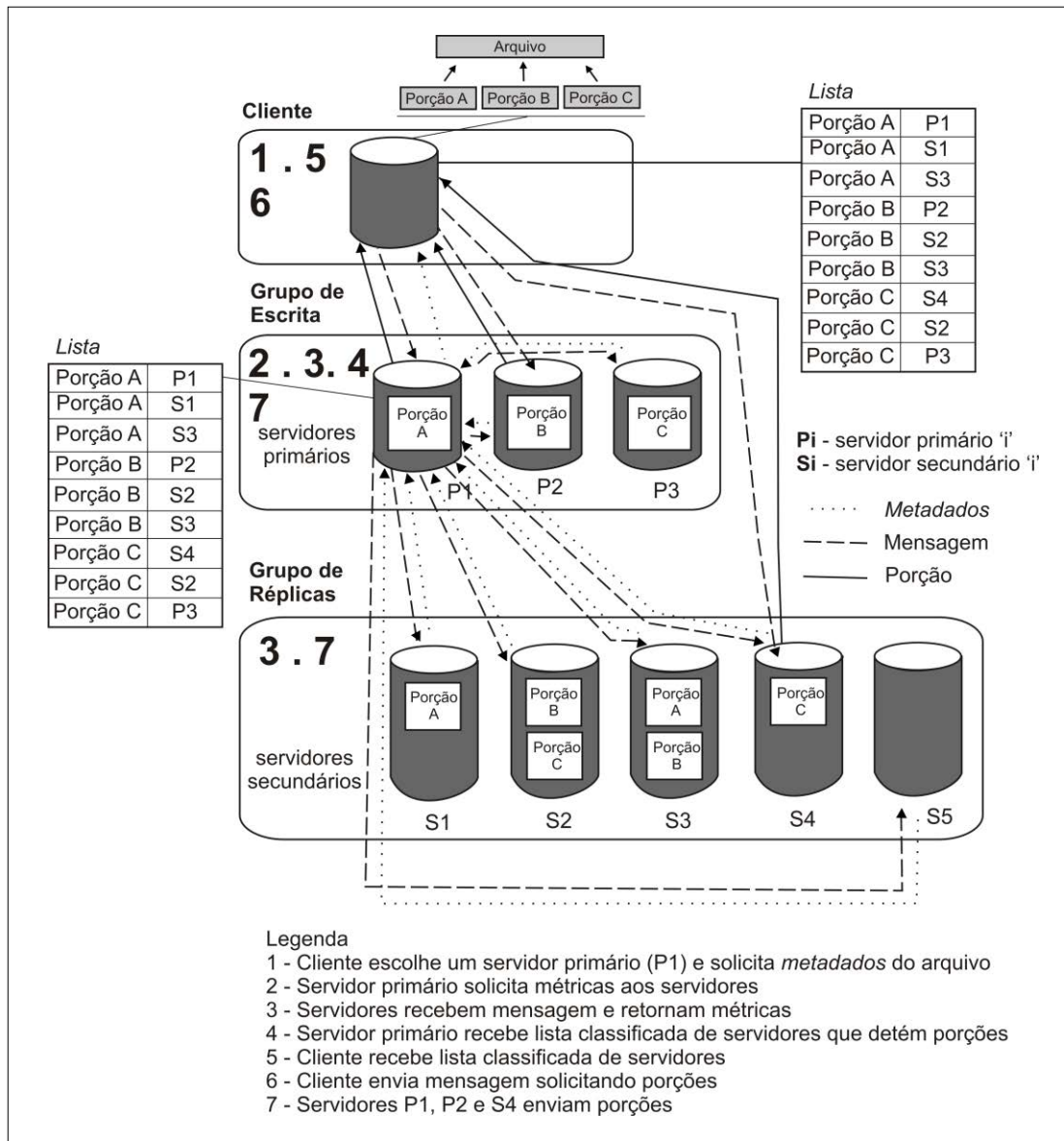
Com base na lista classificatória, a estação cliente requisita as porções que compõem o arquivo aos servidores melhores classificados que detêm as porções. Para fazer uso do arquivo, o cliente junta as porções do arquivo recebidas e descriptografa o arquivo.

Na Figura 19, é representada a operação de leitura no FlexA desenvolvido como exemplo.

Neste exemplo, inicialmente o cliente escolhe um dos servidores primários ativos de forma aleatória (neste exemplo, o servidor primário P1) e envia mensagem solicitando os *metadados* do arquivo que deseja efetuar a leitura. O servidor primário P1 solicita a todos os servidores suas métricas, monta uma lista ordenada (*metadados*) e encaminha esta lista ordenada (Lista) ao cliente.

O cliente, ao receber a lista ordenada, faz a solicitação das porções que compõem o arquivo aos servidores melhores classificados para cada porção. Neste exemplo, a ‘Porção A’ é solicitada para o servidor P1, a ‘Porção B’ é solicitada para o servidor P2 e a ‘Porção C’ é solicitada para o servidor S4. Efetuada a transferência das porções para o cliente, este faz a junção dessas porções, descriptografa o arquivo e torna o arquivo disponível ao usuário.

Figura 19 – Operação de leitura no FlexA desenvolvido



4.4 Autoavaliação do servidor primário

No FlexA desenvolvido, a queda de uma estação que faz parte do sistema de arquivos distribuído tem diferentes impactos, dependendo do tipo de estação em questão.

Se um cliente fica indisponível, sua recuperação é deixada para o usuário: arquivos que foram abertos em um cliente são considerados perdidos e o sistema mantém o arquivo armazenado nos servidores com a última atualização enviada.

Por outro lado, se um servidor secundário torna-se indisponível, na operação de escrita esta estação automaticamente não estará disponível para utilização, já que existe um *daemon* que executa em segundo plano nos clientes, e servidores identificam essa ausência. No caso

da operação de leitura, o cliente solicita à outra estação do tipo servidor secundário disponível as porções, já que o servidor secundário que apresentou a falha, não estará disponível na lista de porções gerada quando da operação de leitura.

Os servidores primários, por outro lado, demandam um tratamento especial no caso de apresentarem indisponibilidade. Este tratamento especial torna-se necessário uma vez que estes servidores são encarregados de manter os *metadados* do arquivo que são informados quando da operação de leitura aos clientes, pelo fato de iniciarem a operação de replicação de arquivos na operação de escrita e por armazenarem porções na operação de escrita.

No sentido de garantir que pelo menos três servidores primários estejam disponíveis no sistema, as modificações no sentido de agregar melhor disponibilidade no sistema FlexA original passou a contar com a operação de autoavaliação do servidor primário. Dessa forma, caso um servidor primário fique indisponível, um servidor secundário é escolhido para desempenhar a função de servidor primário, por meio da inicialização do `Coletor` – servidor primário. A autoavaliação de um servidor primário é realizada assim que o sistema é iniciado, através das fases de detecção, eleição e substituição, abordadas a seguir.

Detecção - A detecção da queda de um servidor primário é feita por um mecanismo de sondagem que é realizado a cada segundo. Ao iniciar o `Coletor` – servidor primário, cada servidor primário ativo cria uma lista contendo os servidores primários que fazem parte do sistema. Como exemplo, na Figura 20 (A) é apresentada lista gerada no servidor primário com IP 192.168.10.10, em que os servidores primários 192.168.10.10, 192.168.10.20 e 192.168.10.30 estão ativos no sistema (o IP do servidor primário que gerou a lista também é mencionado).

Figura 20 – Servidores primários esperados, servidores ativos e desativados

192.168.10.10	192.168.10.20	192.168.10.30	(A)
192.168.10.10	<i>null</i>	192.168.10.30	(B)
192.168.10.20			(C)

De posse dessa lista, o servidor primário passa a verificar se cada um dos servidores constantes na lista continua ativo, montando uma nova lista de servidores. Caso algum dos

servidores previstos não responda, o item da lista correspondente ao servidor primário que não respondeu é modificado: ao invés do endereço IP, será inserido o valor “*null*”. Como exemplo, na Figura 20 (B) é apresentada nova lista de servidores tendo o valor “*null*” para o servidor com o endereço IP 192.168.10.20 pelo fato deste não ter respondido à solicitação no momento da verificação.

A lista dos servidores primários que fazem parte do sistema é confrontada com a nova lista que foi gerada a partir da sondagem dos servidores primários, gerando uma nova lista contendo o endereço IP das estações que não responderam. Neste exemplo, a Figura 20 (C) apresenta lista contendo o endereço IP do servidor primário que não respondeu (192.168.10.20).

Após a verificação de que um dos servidores primários não está ativo, a sondagem é interrompida nos servidores primários. Nesse instante, cada servidor primário ativo envia aos seus pares a lista apresentada na Figura 22 (C), ou seja, o servidor 192.168.10.10 envia mensagem contendo a lista com o IP 192.168.10.20 para o servidor com IP 192.168.10.30 e vice-versa.

Ao receber a lista contendo o servidor primário inativo (192.168.10.20), cada servidor primário aguarda dez segundos e faz uma nova verificação para certificar de que realmente o servidor primário está inativo. Em caso negativo, a sondagem é reestabelecida.

No entanto, caso o servidor realmente esteja inativo, torna-se necessária a eleição de um servidor secundário para assumir o lugar do servidor primário que se tornou indisponível.

A Figura 21 apresenta a fase ‘detecção’, que acontece em duas etapas. Na etapa A, o servidor com IP 192.168.10.10 carrega lista que contém todos os servidores primários e inicia a verificação. Na etapa B, ao detectar a ausência do servidor primário com IP 192.168.10.20, este servidor envia a seus pares a lista contendo a estação que não respondeu à solicitação para que estes servidores possam confirmar a ausência do servidor.

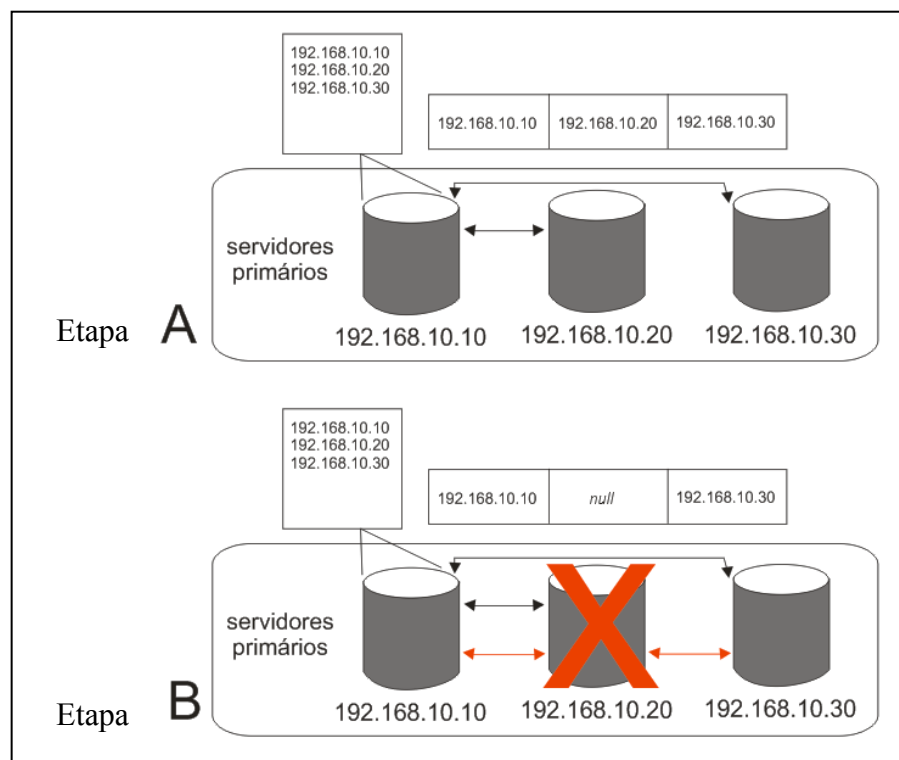
Eleição – Nesta fase, um dos servidores primários remanescentes escolhe de forma aleatória um servidor secundário disponível para liderar a eleição e envia uma mensagem para que este servidor inicie a eleição de um servidor secundário para assumir o lugar do servidor primário indisponível.

Pelo fato dos processos ativos nos servidores secundários estarem organizados em anel lógico e existir um canal de comunicação comum entre todos os servidores, para que a eleição seja concretizada, o líder solicita informações aos servidores secundários ativos considerando o espaço em disco (capacidade de armazenar arquivos) e a ocupação do canal de comunicação

(capacidade de responder rapidamente a requisições) adotando-se métricas mencionadas na Figura 17.

A eleição é concluída após a coleta de informações de todos os servidores secundários e o retorno do IP do servidor eleito para o servidor secundário que iniciou a eleição. Neste momento, o servidor secundário que conduziu a eleição envia o endereço IP do servidor secundário eleito ao servidor primário.

Figura 21 – Fase ‘detecção’ na avaliação de servidores primários



Substituição – Nesta última fase, o servidor primário, ao receber o endereço IP do servidor secundário que assumirá as funções de servidor primário, inicia as etapas para que o sistema possa voltar a trabalhar.

A primeira etapa é enviar mensagem ao servidor secundário escolhido notificando-o de que foi escolhido para atuar como servidor primário na ausência de um servidor primário.

Ao receber a mensagem, o servidor secundário realiza a sincronização da base de dados junto a um dos servidores primários ativos. Nesta fase, o servidor secundário, que passará a exercer o papel de servidor primário, necessita das informações de localização dos arquivos; assim, a base de dados do servidor primário é enviada ao servidor secundário. Nesta operação é utilizado o algoritmo *hash* de 128 bits MD5 (*Message-Digest Algorithm 5*) (RIVEST, 1992), para garantir a integridade do arquivo trocado entre os servidores.

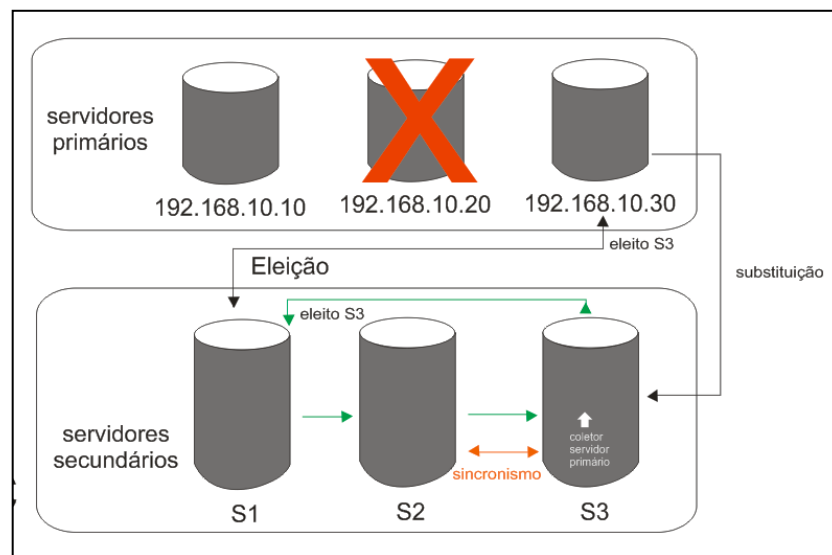
Realizada a sincronização, o servidor secundário eleito para exercer atividades de servidor primário inicia o módulo `Coletor` - servidor primário e torna-se ativo para atuar como um servidor primário. É importante ressaltar que este servidor secundário eleito continuará a desempenhar as atividades de servidor secundário e acumulará a função de servidor primário.

Nos clientes, um *daemon* que verifica a presença de novos servidores identifica o novo servidor primário, ficando disponível para oferecer *metadados* de arquivos no processo de leitura e receber porções de arquivos no processo de escrita.

Nos servidores primários e servidores secundários, um *daemon* identifica o servidor primário e este passa a fazer parte do sistema nas operações de escrita e leitura.

No caso do servidor primário que apresentou falha voltar à atividade, um *daemon* nos clientes e servidores localiza este servidor primário, atualizando a lista de servidores primários ativos. O servidor primário efetua a sincronização dos *metadados*, passando a atuar nas operações de escrita e leitura. A Figura 22 apresenta as fases ‘eleição’ e ‘substituição’.

Figura 22 - Fases ‘eleição’ e ‘substituição’ na avaliação de servidores primários



Na Figura 22 apresentada, o servidor com IP 192.168.10.30 envia mensagem ao servidor secundário S1 (escolhido de forma aleatória), para que inicie a eleição de um servidor secundário. Após percorrer a rede solicitando as métricas dos servidores secundários, o servidor S3 é eleito, tendo essa informação enviada ao servidor 192.168.10.30, que solicitou o início de uma eleição. Tem-se início a fase de substituição, com o servidor 192.168.10.30 enviando mensagem ao servidor eleito S3. O servidor S3, ao receber a mensagem, realiza o

sincronismo da base de dados e a ativação do `Coletor` - servidor primário, concretizando a fase de substituição.

4.5 Avaliação da sobrecarga do servidor primário

Outro item agregado neste trabalho no sentido de melhorar a disponibilidade do FlexA original, foi a avaliação da sobrecarga dos servidores primários que compõem o sistema. Dessa forma, ao detectar a sobrecarga do servidor primário, um servidor secundário passa a atuar também como servidor primário, após iniciar o `Coletor` – servidor primário.

Pelo fato dos servidores primários armazenarem os *metadados* dos arquivos (utilizado nas operações de escrita e leitura), e serem responsáveis por armazenar porções dos arquivos constantes no sistema, é de extrema importância que estes servidores sejam monitorados quanto a sua sobrecarga, no sentido de não exceder sua capacidade e conseqüentemente tornar o sistema indisponível. Neste sentido, determinou-se que os recursos a serem avaliados para determinar a sobrecarga do servidor primário são: porcentagem de disco utilizado, porcentagem de atividade do disco, porcentagem de memória utilizada e porcentagem de operação da rede.

As informações sobre o uso desses recursos são utilizadas para compor um índice de disponibilidade do servidor primário, tendo a porcentagem relacionada a um peso, conforme apresentado na Tabela 1.

O processo de avaliação de sobrecarga do servidor primário é composto de cinco etapas: coleta das informações, classificação da situação do servidor, análise das classificações, eleição e habilitação do servidor secundário como servidor primário (caso seja necessário).

Tabela 1 – Métricas para compor índice de disponibilidade do servidor primário

Recurso	Pontuação		
	0% a 50%	51% a 75%	76% a 100%
Disco utilizado	0,5	1	3
Atividade do disco	0,5	1	3
Memória utilizada	0,5	1	2
Operação de rede	1	1	3

Coleta das informações - A coleta das informações é realizada por cada um dos servidores primários após o início do módulo Coletor - servidor primário, quando o sistema é carregado. Após a coleta das informações, é calculado o índice de disponibilidade com o uso da equação apresentada na Figura 23.

Figura 23 – Índice de disponibilidade na sobrecarga do servidor primário

$$\text{Índice de Disponibilidade} = \left(\text{DU} + \text{AD} + \text{MU} + \text{OR} \right)$$

DU = Disco utilizado
AD = Atividade disco
MU = Memória utilizada
OR = Operação da rede

Classificação da situação do servidor - Após o cálculo do índice de disponibilidade, é gerada a classificação do servidor primário, dependendo da pontuação obtida: caso o índice seja menor que 5, o servidor primário é classificado como ‘normal’, caso contrário, se for igual ou maior que 5, é classificado como ‘sobrecarregado’.

A cada 10 minutos, a coleta é realizada e o sistema é classificado, tendo esta classificação inserida em uma lista que contém as classificações realizadas. Na Figura 24 (A) é apresentada uma lista contendo as duas primeiras classificações como ‘sobrecarregado’.

Figura 24 – Classificação na sobrecarga do servidor primário

Classificação	sobrecarregado	sobrecarregado	(A)
	Coleta inicial	Após 10 minutos	
Classificação	sobrecarregado	sobrecarregado	sobrecarregado (B)
Classificação	sobrecarregado	sobrecarregado	normal (C)

Análise das classificações - Quando a lista de classificações completa três entradas, esta passa por uma análise: caso estejam presentes três classificações ‘sobrecarregado’, o sistema é considerado sobrecarregado, a fase de coleta de informações é interrompida e inicia-

se a próxima fase: eleição. Na Figura 24 (B) é apresentada a lista com classificações, sendo três entradas ‘sobrecarregado’, indicando o início da próxima fase.

Por outro lado, o sistema pode ser considerado normal caso exista pelo menos uma classificação ‘normal’ na lista que contém as classificações. Na Figura 24 (C) é apresentada lista com classificações contendo uma classificação ‘normal’; neste caso o sistema não é considerado sobrecarregado e a próxima fase, de ‘eleição’, não terá início.

A fase de coleta de informações e classificação continua de dez em dez minutos, sempre analisando a lista de classificação com as três últimas classificações, deslocando a lista de classificações para a esquerda, na entrada de um novo item.

Eleição – Caso o servidor primário seja considerado sobrecarregado, cabe ao servidor primário escolher aleatoriamente um dos servidores secundários ativos e enviar uma mensagem para que este servidor inicie a eleição de um servidor secundário que atuará também como servidor primário.

A eleição utiliza algoritmo baseado em anel (COULOURIS; DOLLIMORE; KINDBERG, 2007; MOLINA, 1982), iniciando-se no servidor secundário escolhido para liderar a eleição.

São solicitados o espaço em disco e a ocupação do canal de comunicação dos servidores secundários considerando a mesma equação apresentada na Figura 17. Uma vez coletadas as informações desses servidores, o líder recebe o IP do servidor eleito, retornando este IP para o servidor primário que solicitou a eleição.

Habilitação do servidor secundário como servidor primário - Após a definição do servidor secundário que se tornará servidor primário, sua habilitação consiste no envio, a partir do servidor primário, de mensagem solicitando que o servidor secundário inicie o módulo `Coletor` – servidor primário.

Ao receber a mensagem, o servidor secundário inicialmente solicita a um dos servidores primários os *metadados* dos arquivos que estão armazenados e os armazena em seu banco de dados; essa operação é chamada de sincronização. Após a sincronização, o servidor secundário eleito inicia o `Coletor` – servidor primário, passando a atuar também como servidor primário. Cabe ressaltar que este servidor manterá o `Coletor` – servidor secundário ativado, atuando agora como servidor primário e servidor secundário.

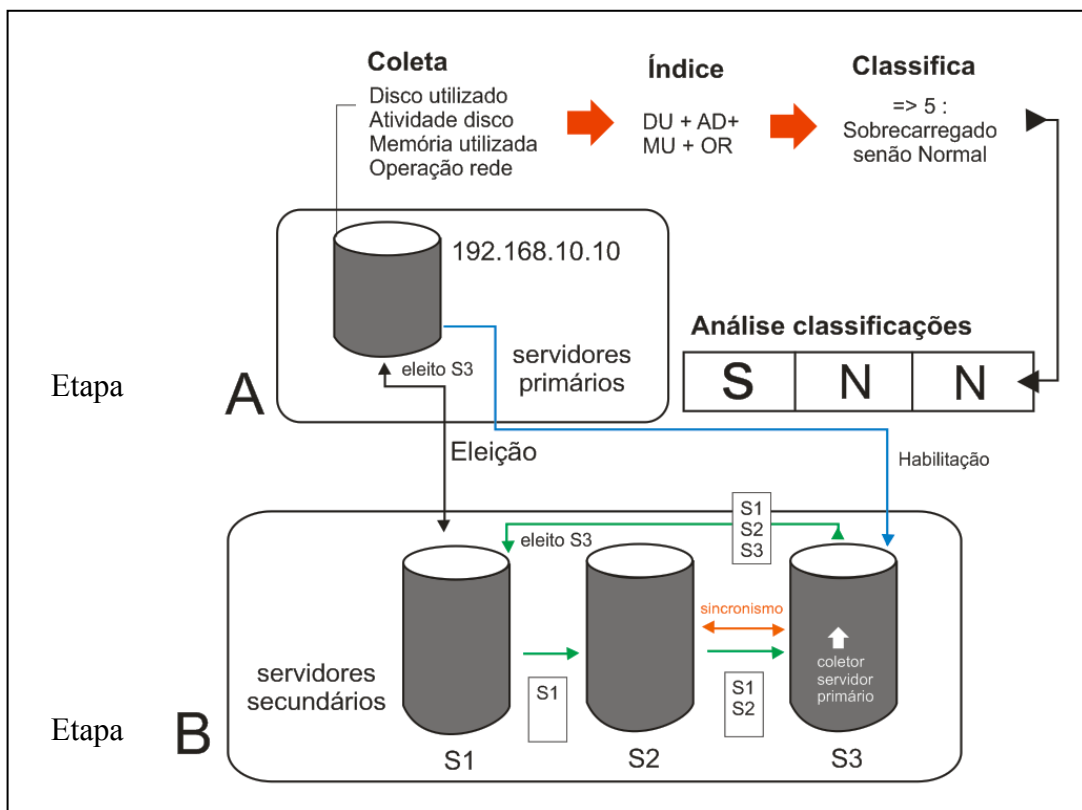
Um *daemon* presente nos clientes, servidores primários e servidores secundários identificam este novo servidor primário no sistema, que passa a fazer parte dos servidores primários disponíveis nas operações de escrita e leitura de arquivos.

O servidor primário que apresentou a sobrecarga passa a não fazer parte da lista de servidores primários disponíveis pelo fato de ter sido interrompido seu `Coletor` - servidor primário quando da identificação da sobrecarga na fase de análise.

A coleta de informações continua e, caso o servidor volte a apresentar estado normal, o `Coletor` - servidor primário é iniciado. Neste caso, o sistema passará a contar com quatro servidores primários ativos; sendo que na operação de escrita, três servidores serão escolhidos para armazenar as porções de um arquivo que for submetido ao sistema.

Na Figura 25 é apresentado o processo de avaliação da sobrecarga do servidor primário.

Figura 25 – Avaliação da sobrecarga do servidor primário



Na etapa A da Figura 25, o servidor primário com IP 192.168.10.10 inicia a autoavaliação logo após ser iniciado, realizando a coleta das informações (disco utilizado, atividade em disco, memória utilizada e porcentagem de operação da rede). Após a coleta é calculado o índice de disponibilidade e o sistema classifica o índice de disponibilidade. Após

três coletas, uma lista é gerada (na figura, ‘S’, N’ e ‘N’, refletindo em uma classificação ‘sobrecarregado’ e duas ‘normal’), culminando na análise das classificações.

No caso de sobrecarga do servidor primário, a etapa B se inicia com o envio, a partir do servidor primário sobrecarregado (192.168.10.10), de mensagem para que um servidor secundário (S1) inicie a eleição do servidor secundário que assumirá como primário. Após a eleição, o servidor primário (192.168.10.10) recebe uma mensagem contendo o IP do servidor secundário escolhido (S3) e logo após o recebimento envia mensagem para o servidor escolhido (S3) para que inicie a fase ‘habilitação’. A fase de ‘habilitação’ contempla o processo através do recebimento de uma mensagem por parte do servidor S3, eleito, para que sincronize a base de dados e inicie o Coletor - servidor primário.

4.6 Avaliação da sobrecarga do servidor secundário

Com a habilitação do Grupo de Réplicas, o FlexA desenvolvido passou a contar com os servidores secundários. No entanto, como esses servidores armazenam porções que compõem um arquivo, é importante que sejam monitorados quanto a sua sobrecarga no sentido de não exceder sua capacidade. Esse monitoramento é denominado avaliação da sobrecarga do servidor secundário que, uma vez identificado, habilita um cliente para substituir o servidor secundário sobrecarregado.

Neste sentido, determinou-se que os recursos a serem avaliados para determinar a sobrecarga do servidor secundário são: porcentagem de disco utilizado e tendência de utilização do disco. A tendência de ocupação pode assumir dois valores: ‘aumentando’ ou ‘diminuindo/estável’. Para determinar este conceito, são verificadas a utilização de disco no tempo t' e t'' ; caso a utilização do disco seja maior no tempo t'' , o conceito é classificado como ‘aumentando’, pelo fato do espaço de ocupação em disco estar aumentando. Caso contrário, o conceito é determinado como ‘diminuindo/estável’.

As informações sobre o uso desses recursos são usadas para compor índice de disponibilidade do servidor secundário, tendo a porcentagem relacionada a um peso, conforme apresentado na Tabela 2.

O processo de avaliação de sobrecarga do servidor secundário é composto de cinco etapas: coleta das informações, classificação da situação do servidor, análise das classificações, eleição e habilitação de cliente como servidor secundário (caso seja necessário).

Tabela 2 – Métricas para compor índice de disponibilidade do servidor secundário

Recurso	Pontuação	
	0% a 75%	76% a 100%
Disco utilizado	1	2
Tendência de ocupação	Aumentando	Diminuindo/Estável
	1	0

Coleta das informações - A coleta das informações é feita por cada um dos servidores secundários após o início do módulo Coletor - servidor secundário. Após a coleta, caso o valor ‘Disco utilizado’ seja igual ou maior que 76%, o valor 2 é assumido; para valores menores, assume-se o valor 1. Para definir a tendência de ocupação em disco, caso a tendência de ocupação esteja aumentando, assume-se o valor 1; caso contrário, o valor 0.

Uma vez coletadas as informações do sistema e associadas a uma pontuação, o servidor secundário calcula o índice de disponibilidade por meio da equação apresentada na Figura 26.

A equação para determinar o índice de disponibilidade do servidor secundário é diferente do índice apresentado anteriormente, que calcula a indisponibilidade do servidor primário.

Na equação do servidor secundário, são utilizadas informações da porcentagem de disco utilizado e da tendência de ocupação em disco, uma vez que o uso de memória e rede é menor, se comparado a um servidor primário.

Figura 26 - Índice de disponibilidade na sobrecarga do servidor secundário

$$\text{Índice de Disponibilidade servidor secundário} = (DU + TD)$$

DU = Disco utilizado
TD = Tendência ocupação disco

Classificação da situação do servidor - Após a definição do índice de disponibilidade do servidor secundário, é gerada uma classificação, dependendo da pontuação

obtida: caso o índice de disponibilidade seja maior ou igual a 3, o servidor secundário é classificado como ‘sobrecarregado’; caso contrário, classificado como ‘normal’.

Após a primeira coleta, o sistema aguarda 10 minutos para gerar nova coleta e classificação.

Na Figura 27 é apresentada uma lista classificatória contendo as duas primeiras classificações, sendo a primeira classificação ‘sobrecarregado’ e a segunda classificação ‘normal’.

Figura 27 – Lista classificatória com classificações diferentes

Classificação	sobrecarregado	normal
	Coleta inicial	Após 10 minutos

Análise das classificações - Quando a lista com as classificações do servidor secundário completa três entradas, esta passa por uma análise: caso estejam presentes três classificações ‘sobrecarregado’, o sistema é considerado sobrecarregado, a fase de coleta de informações é interrompida e inicia-se a próxima fase, ‘eleição’ - elegendo um novo cliente.

Por outro lado, o sistema pode ser considerado estável caso exista pelo menos uma classificação ‘normal’ na lista de classificação. Neste caso, a coleta de informações e classificação continua de dez em dez minutos, sempre mantendo a lista de classificação com as três últimas classificações, deslocando a lista de classificações para a esquerda, na entrada de um novo item.

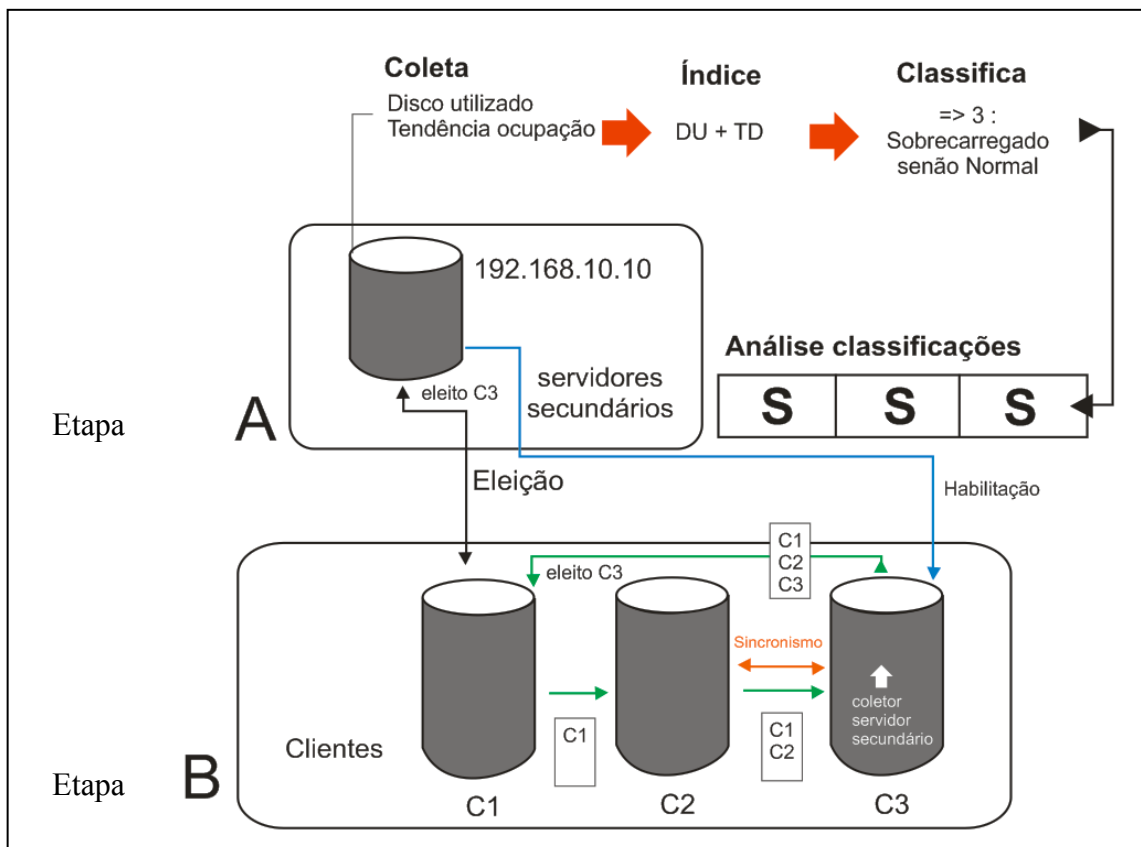
Eleição – Caso o servidor secundário seja considerado sobrecarregado após transcurso das fases apresentadas anteriormente, o servidor secundário escolhe aleatoriamente um dos clientes ativos e envia uma mensagem para que este cliente inicie a eleição de um cliente que atuará como servidor secundário, pela eleição baseada em anel, conforme explicado na seção 4.5.

São solicitados o espaço em disco e a ocupação do canal de comunicação dos clientes, considerando a mesma equação apresentada na Figura 17. Uma vez coletadas as informações desses clientes, o líder recebe o IP do cliente eleito, retornando este IP ao servidor secundário.

Habilitação do cliente como servidor secundário – A habilitação do cliente consiste no envio, a partir do servidor secundário, de mensagem solicitando que o cliente eleito inicie o módulo `Coletor - servidor secundário`. Após receber a mensagem, o cliente sincroniza a base de dados e inicia o `Coletor - servidor secundário`, passando a atuar também como servidor secundário. É importante ressaltar que o cliente, além de desempenhar as operações de escrita e leitura, passa a desempenhar também as funções de um servidor secundário.

Um *daemon* presente nos clientes, servidores primários e servidores secundários identifica este novo cliente no sistema, que passa a fazer parte do Grupo de Réplicas, disponível para as operações de escrita e leitura de arquivos. O servidor secundário que apresentou a sobrecarga passa a não fazer parte dos servidores disponíveis. A coleta de informações continua e, caso o servidor volte a apresentar estado normal, o `Coletor - servidor secundário` é iniciado. Neste caso, o sistema passará a contar com mais um servidor secundário; na operação de escrita, este servidor poderá ser escolhido para armazenar porções e na operação de leitura, poderá ser contatado caso seja classificado e possua porções que compõem o arquivo requisitado. Na Figura 28 é apresentado o processo de avaliação da sobrecarga do servidor secundário.

Figura 28 – Avaliação da sobrecarga do servidor secundário



As fases ‘coleta’ e ‘classificação’ são apresentadas na etapa A. Na etapa B, caso o servidor secundário seja considerado sobrecarregado, o cliente C1 recebe mensagem (a partir do servidor secundário sobrecarregado) para iniciar a eleição de um cliente que se tornará servidor secundário; o cliente é escolhido com base em métricas e retornado ao servidor secundário. Este servidor envia mensagem ao cliente eleito (neste exemplo, o cliente C3); é realizado o sincronismo da base de dados e iniciado o `Coletor` – servidor secundário.

4.7 Considerações finais

Neste capítulo, foram apresentadas as funcionalidades desenvolvidas no FlexA desenvolvido, abordando as atividades de incorporação do Grupo de Réplicas, autoavaliação do servidor primário, sobrecarga de servidores primários e sobrecarga de servidores secundários com detalhes.

No Capítulo 5 é apresentada a validação do sistema contendo todas as atividades desenvolvidas neste capítulo. No Capítulo 6 são apresentadas avaliações de desempenho e comparações entre o FlexA desenvolvido e os sistemas de arquivos distribuídos FlexA original, Tahoe-LAFS e NFS.

5 VALIDAÇÃO DO SISTEMA

5.1 Considerações iniciais

O processo de validação tem como objetivo analisar a eficácia de todas as mudanças realizadas no FlexA desenvolvido.

Neste capítulo, inicialmente é apresentado o cenário de realização das validações, seguido da apresentação da forma com que o sistema passou a se comportar após as modificações propostas, culminando na validação das operações de escrita e leitura e dos processos de autoavaliação do servidor primário, sobrecarga do servidor primário e sobrecarga do servidor secundário.

5.2 Cenário para validação

A validação foi realizada utilizando o *cluster* instalado no laboratório do GSPD, que é composto por computadores com diferentes configurações, divididas em dois grupos:

Grupo A - computadores com processador Intel Core i7-3770 CPU 3,40GHz de 4 núcleos e 8 *threads*, Memória RAM de 16GB, HDD de 500GB 7200RPM SATA II, Rede *Gigabit Ethernet* (1000 Mbps) e sistema operacional Debian GNU/Linux 7.1.0 (Linux *kernel* versão 3.2.0-4-amd64).

Grupo B – computadores com processador Intel Pentium DualCore CPU E2160 1,8Ghz, Memória RAM de 2GB, HDD de 40GB 7200RPM IDE, Rede *Gigabit Ethernet* (1000 Mbps) e sistema operacional Debian GNU/Linux 7.1.0 (Linux *kernel* versão 3.2.0-4-amd64).

A avaliação foi efetuada em um cenário composto por 8 computadores no Grupo A e 8 computadores no Grupo B, variando entre servidores e clientes.

O grupo A, com melhor *hardware*, recebeu os clientes, já que no FlexA original e no FlexA desenvolvido a maior parte do processamento fica a cargo dos clientes. O grupo B ficou responsável por receber os servidores.

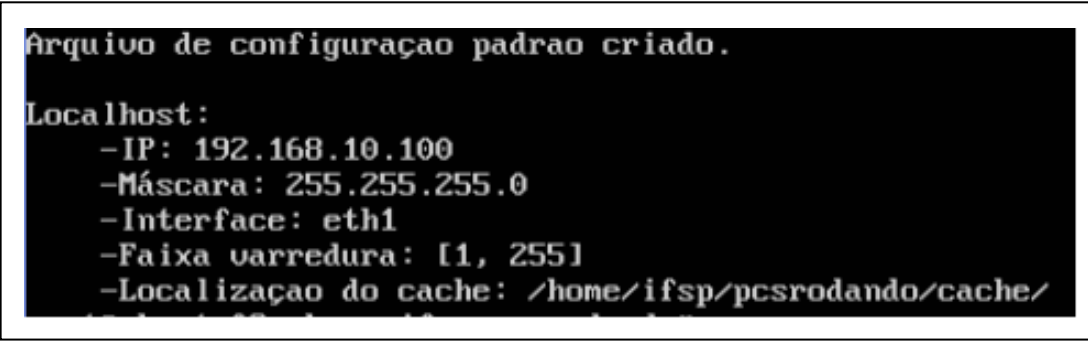
Visando dispor de um maior número de clientes para a realização dos testes, utilizou-se o conceito de virtualização. Dessa forma, cada computador do grupo A recebeu duas máquinas virtualizadas operando com dois clientes com a configuração: um processador, Memória RAM de 1GB, HDD de 10GB, Rede *Gigabit Ethernet* (1000 Mbps) e sistema operacional Debian GNU/Linux 7.1.0 (Linux *kernel* versão 3.2.0-4-amd64).

5.3 Funcionamento do sistema

Após a instalação do sistema de arquivos distribuído FlexA desenvolvido, descrita no Apêndice A, um arquivo de configuração (`config.dat`) é gerado contendo as informações: endereço IP, máscara da rede, interface de rede, faixa de varredura e localização da *cache*. Por padrão, o diretório *cache* é criado no diretório corrente, onde o FlexA desenvolvido foi executado.

Na Figura 29 são apresentados dados do arquivo de configuração para uma estação cliente (IP 192.168.10.100).

Figura 29 – Informações de configuração no cliente



```
Arquivo de configuração padrao criado.  
Localhost:  
-IP: 192.168.10.100  
-Máscara: 255.255.255.0  
-Interface: eth1  
-Faixa varredura: [1, 255]  
-Localização do cache: /home/ifsp/pcsrodando/cache/
```

A interação com o sistema FlexA desenvolvido é feita através do *prompt* de comandos, assim como na versão do FlexA original. No Quadro 2 são apresentados os comandos do FlexA desenvolvido, tendo-se mantidos os comandos do FlexA original e adicionados os comandos do FlexA desenvolvido mediante as modificações propostas.

Quadro 2 – Comandos do FlexA desenvolvido

Comando	Opção	Descrição
com.py	-r	Identifica a rede, gerando arquivo de configuração
com.py	-b	Busca servidores primários na rede
com.py	-db	Busca servidores primários de forma contínua
com.py	-s	Busca servidores secundários na rede
com.py	-ds	Busca servidores secundários de forma contínua
com.py	-c	Busca clientes na rede
coletor_servidor.py		Inicia módulo coletor primário
coletor_secundario.py		Inicia módulo coletor secundário
coletor_cliente.py		Inicia módulo coletor cliente
flexa.py	-p <arquivo>	Escreve um arquivo no FlexA desenvolvido
flexa.py	-g <arquivo>	Lê um arquivo no FlexA desenvolvido
flexa.py	-l <arquivo, *>	Lista arquivo específico ou todos os disponíveis
flexa.py	-d <arquivo>	Apaga um arquivo no FlexA desenvolvido
flexa.py	-np<arquivo>	Gera novas permissões do arquivo

Após o início do sistema FlexA desenvolvido, os servidores primários iniciam a operação de ‘autoavaliação do servidor primário’, para garantir que a ativação de um novo servidor primário seja realizada na indisponibilidade de um servidor deste tipo. Na Figura 30 é apresentada tela com o início da autoavaliação de um servidor primário disponível no sistema.

Figura 30 – Início da autoavaliação no servidor primário

```

*****
SONDAGEM de PRIMARIOS ativos
*****
IP: 192.168.10.101
IP: 192.168.10.102
IP: 192.168.10.103

```

5.3.1 Operação de escrita

A operação de escrita inicia-se no cliente com a criptografia e a divisão do arquivo em três porções. Neste exemplo foram utilizados um cliente (IP 192.168.10.100), três servidores

primários (IPs 192.168.10.101, 192.168.10.102 e 192.168.10.103) e dois servidores secundários (IPs 192.168.10.104 e 192.168.10.105). Após a divisão do arquivo, são gerados manipuladores de escrita-leitura e somente-leitura no cliente contendo o nome do arquivo, chaves de escrita-leitura e somente-leitura. Como exemplo, na Figura 31 é apresentada a operação de escrita a partir do terminal cliente para a escrita do arquivo flexa_10MB (enviado a partir do comando `flexa.py -p flexa_10MB`), assim como a confirmação de que as porções foram enviadas.

Figura 31 – Operação de escrita no cliente – envio de porções

```
Arquivo particionado em 3 parte(s)
tamanho das porcoes 4674445, 4674445, 4674445
Registrando no Banco de Dados
localporcao 1@192.168.10.102@5000$2@192.168.10.103@5000$3@192.168.10.101@5000$
servidores_porcao. ['192.168.10.102', '192.168.10.103', '192.168.10.101']
servidores_metadados []
Enviando o chunk flexa_10MB.enc-1 para o servidor 192.168.10.102:5000
Enviando o chunk flexa_10MB.enc-2 para o servidor 192.168.10.103:5000
Enviando o chunk flexa_10MB.enc-3 para o servidor 192.168.10.101:5000
wait_file on
Aguardando transferência...
Porção flexa10MB.enc1 transferida para Servidor 192.168.10.102
Porção flexa10MB.enc2 transferida para Servidor 192.168.10.103
Porção flexa10MB.enc3 transferida para Servidor 192.168.10.101
```

No lado dos servidores primários, a saída do módulo `Coletor` - servidor primário exibe o tipo de operação que está sendo realizada sobre o arquivo (escrita, neste caso) e o cliente (IP 192.168.10.100) que solicitou a operação, como apresentado na Figura 32.

Figura 32 – Operação de escrita no cliente – transferência do arquivo

```
Recebendo 497 bytes de cabeçalho
HEADER: [<div>cliente#upload#7000#687760c0-b479-11e3-b5d9-0800271bfff4#Flexa_10M
B.enc-2#4674445#978d9a6c3bb276ff9e7f40d9ffe717a0a91c28abb3cf11c3f6ab558c40380aaf
02f957a7d6027af1fc77488f1b279ea1#0#Flexa_10MB#root/#Tue Mar 25 20:58:45 2014#Tru
e#fb6304868c1fcbb9aea3db2577e2bc3fb1213a4ecb32587f2f450f7df8c6ba52#6e55b1267bd5b
e0a7b34e7d4f7d7d2fc3b741789#1097b449a821d95c96a91ec781cd5abb906fee11bc2cdd1d7fe4
3cbd233124c0#.enc-2#1@192.168.10.102@5000$2@192.168.10.101@5000$3@192.168.10.103
@5000$#192.168.10.103</div>]
Tamanho header 497 bytes
Upload arquivo [Flexa_10MB:4674445 bytes] pelo cliente [192.168.10.100:7000]
Transferindo arquivo...
Armazenando arquivo 978d9a6c3bb276ff9e7f40d9ffe717a0a91c28abb3cf11c3f6ab558c4038
0aaf02f957a7d6027af1fc77488f1b279ea1.enc-2
Registrando na tabela arquivos_sad
```

Após o armazenamento das respectivas porções nos servidores primários, tem início em cada servidor primário a identificação dos servidores secundários que receberão cópias das porções, o envio das porções para cada servidor secundário e a sincronização dos *metadados*.

5.3.2 Operação de leitura

A operação de leitura inicia-se no cliente com a verificação da permissão de escrita-leitura associada ao arquivo a ser lido. Posteriormente, o cliente escolhe um servidor primário ativo, solicitando seus *metadados* (IP 192.168.10.103). Na Figura 33 é apresentada saída do terminal cliente na operação de leitura.

Figura 33 – Operação de leitura no cliente – verificação de permissões

```

Permissao de escrita-leitura encontrada para o arquivo [/home/ifsp/pcsrondando/Flexa_10MB]
Definindo as açoes de execucao para usuário com permissoes e arquivo existente
Download dos metadados
Atribuindo IP aleatorio do hosts.dat: 192.168.10.103
HEADER: <div>cliente#download_metadados#8000#0708d3b2-b47b-11e3-b5d9-0800271bff4f#978d9a6c3bb276ff9e7f40d9ffe717a0a91c28abb3cf11c3f6ab558c40380aaf02f957a7d6027af1fc77488f1b279ea1#0</div>
Solicitacao de metadados encaminhada para o servidor 192.168.10.103
Host cliente: 192.168.10.100
192.168.10.102#5000$2@192.168.10.101#5000$3@192.168.10.103#5000$##Flexa_10MB#root/#6e55b1267bd5be0a7b34e7d4f7d7d2fe3b741789#1097b449a821d95c96a91ec781cd5dbb906fee11bc2cdd1d7fe43cbd233124c0#978d9a6c3bb276ff9e7f40d9ffe717a0a91c28abb3cf11c3f6ab558c40380aaf02f957a7d6027af1fc77488f1b279ea1
Opcao de Download: download_metadados
Fazendo download do arquivo

```

Após a obtenção dos *metadados*, referenciando o endereço IP dos servidores que detêm as porções que compõem o arquivo, é feita a solicitação dessas porções aos respectivos servidores. Neste caso, a porção 1 (enc1) é solicitada ao servidor IP 192.168.10.102, porção 2 (enc2) ao servidor IP 192.168.10.101 e porção 3 (enc3) ao servidor IP 192.168.10.103. Neste caso foram escolhidos três servidores primários, com base na equação evidenciada anteriormente. Na Figura 34 é apresentada tela no terminal cliente realizando a solicitação das porções do arquivo aos servidores, aguardando para obtenção das mesmas.

Figura 34 – Solicitação das porções aos servidores

```

Aguardando porcoes...
Solicitacao do chunk [ Flexa_10MB.enc-1 ] encaminhada para o servidor 192.168.10.102
Solicitacao do chunk [ Flexa_10MB.enc-2 ] encaminhada para o servidor 192.168.10.101
Solicitacao do chunk [ Flexa_10MB.enc-3 ] encaminhada para o servidor 192.168.10.103

```

Por outro lado, os servidores escolhidos realizam a validação dos manipuladores do arquivo e enviam a porção ao cliente solicitante.

Na Figura 35 é apresentada tela do servidor primário IP 192.168.10.102 presente no sistema, exibindo o endereço IP do cliente solicitante da porção do arquivo e mensagem informando que a porção foi enviada no processo de leitura para o cliente IP 192.168.10.100.

Figura 35 – Envio de porção de um arquivo a partir de servidor

```
Recebendo 189 bytes de cabeçalho
HEADER: [<diu>cliente#download#7000#0708d3b2-b47b-11e3-b5d9-0800271bff4f#Flexa_1
0MB#.enc-1#978d9a6c3bb276ff9e7f40d9ffe717a0a91c28abb3cf11c3f6ab558c40380aaf02f95
7a7d6027af1fc77488f1b279ea1#root/<diu>]
Tamanho header 189 bytes
Download arquivo [Flexa_10MB.enc-1] para o cliente [192.168.10.100:7000]
Header enviado pelo Upload <diu>download#Flexa_10MB.enc-1#978d9a6c3bb276ff9e7f40
d9ffe717a0a91c28abb3cf11c3f6ab558c40380aaf02f957a7d6027af1fc77488f1b279ea1.enc-1
#4674445#1#root/#<diu>
Arquivo [ Flexa_10MB.enc-1 ] enviado para cliente 192.168.10.100:7000
```

5.3.3 Sobrecarga do servidor primário

A autoavaliação dos servidores primários é realizada por cada servidor primário (no intuito de detectar a sobrecarga do sistema), iniciando-se juntamente com o Coletor - servidor primário. Na Figura 36 é apresentado o primeiro extrato de informações, coletadas no servidor primário. Dentre as informações temos a porcentagem de operação em disco, porcentagem de memória utilizada, porcentagem de disco utilizado e porcentagem de utilização da rede. Para este exemplo foram assumidos um servidor primário que será sobrecarregado (IP 192.168.10.100), e dois servidores secundários (IP 192.168.10.101 e 192.168.10.102).

Figura 36 – Avaliação da sobrecarga de servidor primário

```
#####
EXTRATO DE INFORMACOES - Auto Avaliacao do primário
Operação em disco: 15.87%
Memória utilizada: 49%
HD utilizado: 36.1803029968%
Utilização da rede: maior que 80, 1. Menor, 0 1.0%
#####
Índice operacao disco 0.5
Índice memoria utilizada 0.5
Índice hd utilizado 0.5
Total indices 2.5
Status avaliação Servidor em 2014-03-27 09:01:55.588461 : sobrecarregado
```

Os extratos vão sendo exibidos conforme novas coletas são realizadas. Caso a sobrecarga seja detectada, é exibida no terminal do servidor primário informação de que este servidor está sobrecarregado, são solicitadas as métricas aos servidores secundários presentes

no sistema e efetua-se a eleição para definir o servidor secundário que atuará como servidor primário.

Na Figura 37 é apresentada tela do servidor primário (IP 192.168.10.100) com informação de que está sobrecarregado, sendo solicitadas métricas aos servidores secundários (IP 192.168.10.101 e 192.168.10.102) e exibido o servidor secundário (IP 192.168.10.102) eleito.

Figura 37 – Servidor primário sobrecarregado

```

#####
Primário sobrecarregado - Buscando melhor réplica
-----
Solicitação de metricas encaminhado para a replica 192.168.10.101
Solicitação de metricas encaminhado para a replica 192.168.10.102
Melhor réplica localizada: 192.168.10.102
Pedindo para 192.168.10.102 iniciar coletor_servidor
#####

```

Do lado dos servidores secundários, estes recebem uma solicitação para que enviem suas métricas. Na Figura 38 é apresentada mensagem recebida pelo servidor secundário (IP 192.168.10.101), que devolve métricas para o servidor primário (IP 192.168.10.100).

Figura 38 – Solicitação e devolução de métricas do servidor secundário

```

Sincronizador replica [Ativo] IP: 192.168.10.101; Porta socket: 7500

Recebendo 50 bytes de cabeçalho
Devolvendo as métricas para o computador 192.168.10.100
Metricas enviadas para o servidor 192.168.10.100:7600

```

A operação de autoavaliação do servidor primário é concretizada por pedido para que o servidor secundário escolhido na fase anterior inicie seu Coletor - servidor primário. Na Figura 39 é apresentada tela exibida pelo servidor secundário escolhido (endereço IP 192.168.1.102) recebendo pedido e ativando seu Coletor - servidor primário.

Figura 39 – Servidor secundário inicia Coletor - servidor primário

```

Sincronizador replica [Ativo] IP: 192.168.10.102; Porta socket: 7500
#####
Iniciando Coletor Primario para ajudar na sobrecarga
#####
Sincronizador servidor [Ativo] IP: 192.168.10.102; Porta socket: 5000

```

5.3.4 Sobrecarga do servidor secundário

A avaliação da sobrecarga do servidor secundário é realizada individualmente por cada servidor secundário, sendo que este processo inicia-se juntamente com o Coletor - servidor secundário. Na Figura 40 é apresentado o primeiro extrato de informações coletado no servidor secundário (porcentagem de disco utilizado e tendência de ocupação do disco). Foram assumidos um servidor secundário que fora sobrecarregado (IP 192.168.10.102) e duas estações do tipo cliente.

Figura 40 – Autoavaliação da sobrecarga do servidor secundário

```

Coletando primeira série de informações do servidor de Réplica
////////////////////////////////////
EXTRATO DE INFORMACOES - Auto Avaliacao do primário
HD utilizado: 36.0%
Índice do HD utilizado: 0.5%
Consumo de HD - ESTAVEL ou DIMINUINDO
////////////////////////////////////

```

Os extratos vão sendo exibidos conforme novas coletas são realizadas. Caso a sobrecarga seja detectada, é exibida no terminal do servidor secundário informação de que este servidor está sobrecarregado, apresentado na Figura 41.

Figura 41 – Parando autoavaliação do servidor secundário

```

////////////////////////////////////
Réplica sobrecarregada - Parando autoverificação ...
#####
Réplica sobrecarregada - Buscando melhor cliente

```

A operação de autoavaliação do servidor secundário concretiza-se pelo envio de mensagem para que o cliente, escolhido na fase anterior, inicie seu Coletor - servidor secundário, passando a exercer a função de um servidor deste tipo.

Na Figura 42 é apresentada tela exibida pelo cliente escolhido (endereço IP 192.168.10.202) recebendo pedido e realizando ativação do seu Coletor - servidor secundário.

Figura 42 – Cliente inicia Coletor - servidor secundário



5.4 Validação da autoavaliação do servidor primário

A validação da operação de autoavaliação do servidor primário, realizado em conjunto com OKADA (2013), foi composto por um cenário contendo três servidores primários, quatro servidores secundários e um cliente.

Sobre este cenário foi simulada a queda de um servidor primário (escolhido de forma aleatória) a cada minuto, aguardando 30 segundos antes do início do próximo teste. Este tempo de espera é necessário, uma vez que o servidor primário aguarda dez segundos e efetua um novo teste antes de concluir que o servidor primário realmente está inativo.

Dessa forma, foi verificado se a queda do servidor primário foi detectada e se um servidor secundário foi eleito para assumir o lugar do servidor primário indisponível.

A simulação de queda de um servidor primário foi repetida 200 vezes para verificar se um novo servidor secundário fora eleito. Na Tabela 3 é apresentado o resultado da avaliação, relatando que foram obtidos 100% de êxito de detecção na queda dos servidores primários.

Tabela 3 – Resultado da avaliação de sondagem de servidores primários

Sondagem de servidores	
Número de quedas	200
Quantidade de vezes que a queda foi detectada	200
Porcentagem de sucesso	100%

Após a detecção da queda de um servidor primário, um dos servidores secundários deve iniciar uma eleição para determinar um novo servidor secundário para assumir o lugar do servidor primário inoperante.

Na Tabela 4 é apresentada a quantidade de requisições realizadas (200, devido à simulação de 200 quedas) e a quantidade de eleições que cada servidor secundário realizou.

Como exemplo, verifica-se que o servidor secundário 1 foi escolhido 51 vezes para comandar a eleição.

Tabela 4 – Eleição de servidores secundários

Eleição	
Servidor escolhido para realizar eleição	Quantidade de escolhas
Servidor Secundário 1	51
Servidor Secundário 2	48
Servidor Secundário 3	50
Servidor Secundário 4	51
Total	200
Porcentagem de sucesso	100%

Escolhido o servidor secundário que assumirá como servidor primário, este recebe mensagem para iniciar seu *Coletor* - servidor primário. O novo servidor primário deve atualizar suas informações solicitando *metadados* a um servidor primário ativo, escolhido de forma aleatória. Este processo é denominado ‘sincronização’.

Para finalizar esta etapa, na Tabela 5 são apresentados os tempos médios de sondagem, eleição e sincronização de servidores primários após a realização da avaliação com 200 quedas.

Tabela 5 – Tempos médios das etapas de sondagem, eleição e sincronização

	Sondagem (segundos)	Eleição (segundos)	Sincronização (segundos)	Total (segundos)
Média	13,00773	0,00019	0,00784	13,01577
Desvio Padrão	0,00176	0,00006	0,05466	1,05539

Os tempos de eleição e sincronização são desprezíveis, sendo que o processo de sondagem é o que leva mais tempo. No entanto, é necessário salientar que a sondagem de servidores primários inclui um tempo de espera de 10 segundos para determinar que realmente um servidor primário esteja inativo.

5.5 Validação da avaliação da sobrecarga do servidor primário

A sobrecarga de servidores primários foi validada de duas formas: na primeira, a validação concentrou-se nas operações de escrita e leitura do sistema para analisar a porcentagem do consumo de memória, porcentagem de acesso a disco e a porcentagem de utilização da rede. Em um segundo momento, o sistema foi monitorado no sentido de identificar sua sobrecarga por meio da escrita de arquivos a partir de 16 e 32 clientes.

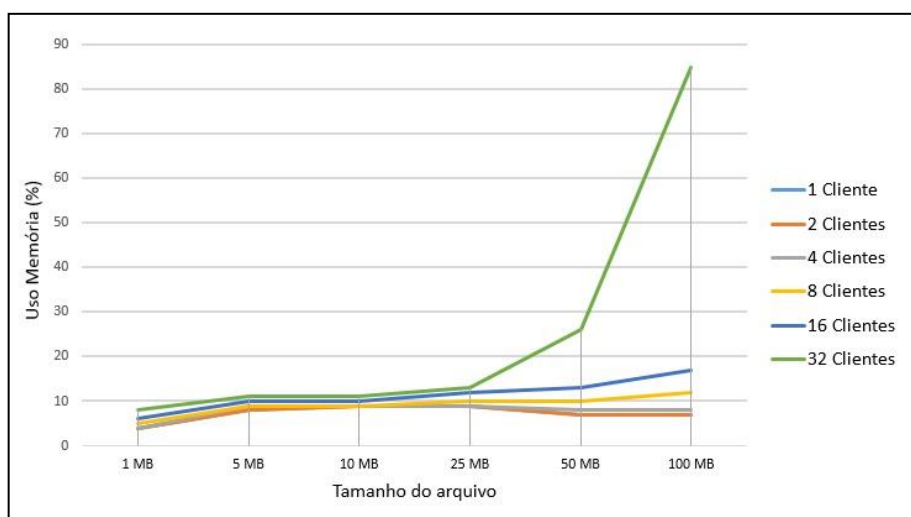
Inicialmente foi abordada a validação das operações de escrita e leitura. O cenário de validação da sobrecarga do servidor primário foi composto de três servidores primários, quatro servidores secundários e trinta e dois clientes com as mesmas configurações relatadas na seção 5.2.

Sobre este cenário foram simuladas as operações de escrita e leitura de arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB com 1, 2, 4, 8, 16 e 32 clientes simultaneamente. No Apêndice B é apresentada tabela contendo a porcentagem do consumo de memória após o envio dos arquivos na operação de escrita.

Quando analisados os tamanhos de arquivos de forma individual, o percentual de uso de memória aumenta conforme o número de clientes e tamanho do arquivo aumentam. Como exemplo temos os arquivos de 50 MB com consumo de 7% para 1 cliente, 10 % para 8 clientes e 26% para 32 clientes. O Gráfico 1 ilustra essas observações.

No entanto, para 4 clientes, o tempo é proporcional até 25 MB; para 50 MB e 100 MB o tempo cai. Para 8, 16 e 32 clientes, o tempo é sempre proporcional: a medida em que aumenta o número de clientes e tamanho do arquivo, o tempo também aumenta.

Gráfico 1 – Consumo de memória na operação de escrita

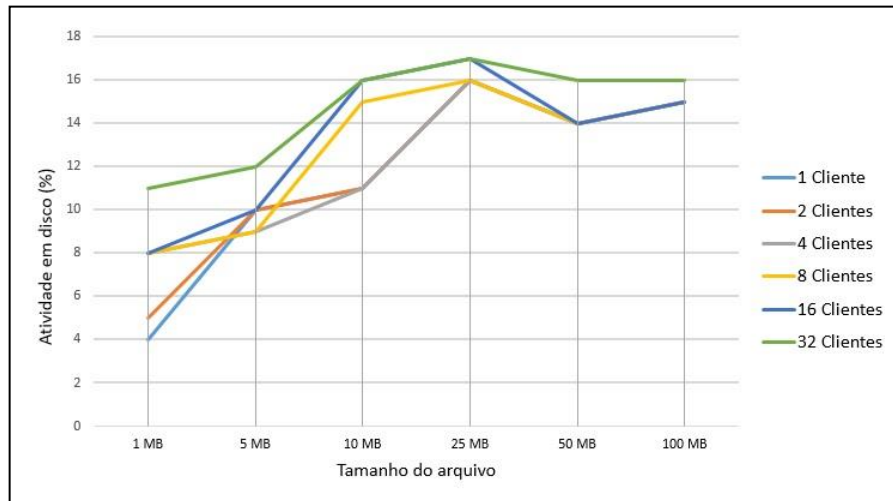


Neste mesmo experimento foi aferida a porcentagem de atividade em disco na operação de escrita, com arquivos do mesmo tamanho e mesma quantidade de clientes. No Apêndice B é apresentada a porcentagem de atividade em disco durante a validação.

A porcentagem de atividade em disco cresce proporcionalmente ao tamanho do arquivo e à quantidade de clientes. No entanto, quando chega em 50 MB, o tempo cai e volta a subir com 100 MB. Esse comportamento pode ser verificado no Gráfico 2.

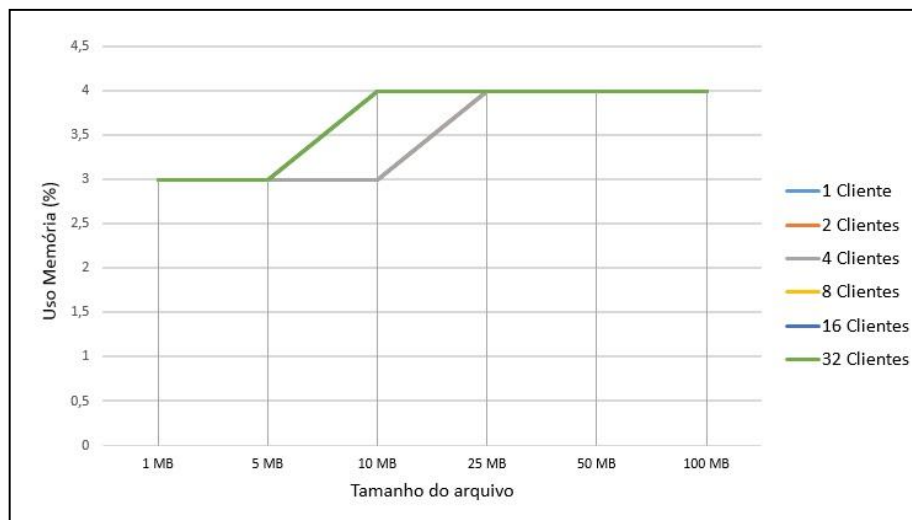
Outro item que foi avaliado foi a porcentagem de atividade da rede. Em todos os casos testados, esta atividade não chegou ao limiar proposto que foi de 75% de utilização da rede, descrito na seção 4.5.

Gráfico 2 – Atividade em disco na operação de escrita



No Apêndice B é apresentada tabela contendo os tempos da validação do comportamento dos servidores primários na operação de leitura. Esta avaliação inicialmente observou a porcentagem do consumo de memória na operação de leitura, apresentado no Gráfico 3.

Gráfico 3 – Consumo de memória na operação de leitura



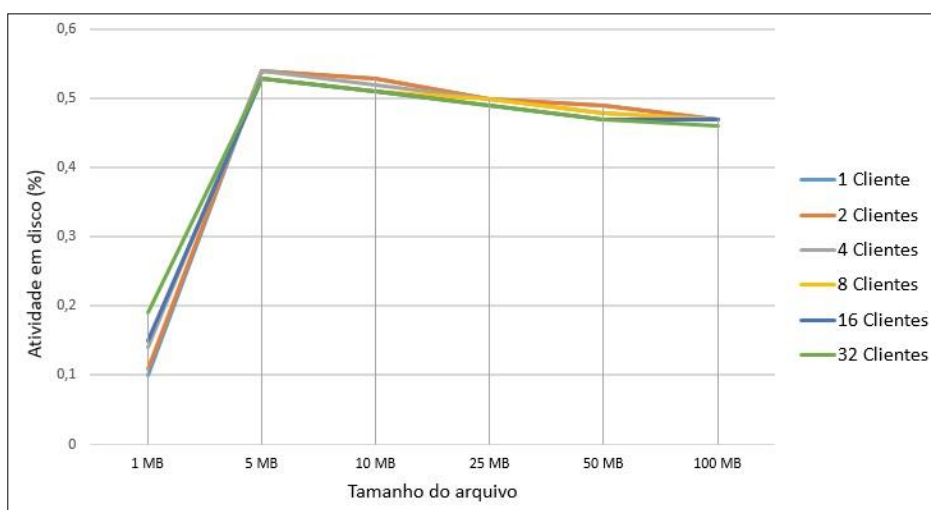
No Gráfico 3, observa-se que o consumo de memória é igual a 3% até 4 clientes enviando arquivos de 10 MB, passando para 4% de utilização para 8 e 16 clientes para arquivos de 10 MB, 25 MB, 50 MB e 100 MB. Dessa forma, o consumo de memória tem um

aumento de 1%, no máximo, para leitura de arquivos na avaliação efetuada. A representação de alguns clientes não pôde ser visualizada no gráfico devido ao fato de que apresentam valores iguais; vide Apêndice B para visualização de todos os dados.

No mesmo experimento foi aferida a porcentagem de atividade em disco na operação de leitura. Quando visto sob a ótica do tamanho do arquivo, a porcentagem aumenta proporcionalmente se comparado a arquivos de tamanhos 1 MB e 5 MB. No entanto, a partir de 10 MB, a porcentagem de atividade passa a cair.

Um limiar semelhante é observado quando é analisado o número de clientes: a partir de 8 clientes, a atividade tende a diminuir. De uma forma geral, quanto maior a sobrecarga do sistema devido ao tamanho do arquivo e quantidade de clientes, menor é a utilização de memória. No Gráfico 4 são apresentados esses dados.

Gráfico 4 – Atividade em disco na operação de leitura



Apresentadas as validações das operações de escrita e leitura, são apresentadas as validações para identificar a sobrecarga do sistema. O cenário de testes foi composto de três servidores primários, quatro servidores secundários e 32 clientes.

Sobre este cenário foi simulada a operação de escrita de arquivos (pelo fato de alterar a porcentagem de ocupação em disco, uso da memória e atividade em disco, que fazem parte da métrica para a definição da sobrecarga do servidor primário) de 50 MB e 100 MB por 16 e 32 clientes. A frequência utilizada foi de vinte vezes para cada operação.

Na Tabela 6 são apresentadas informações da sobrecarga. Nestes testes, o sistema somente foi caracterizado como sobrecarregado no servidor primário 2 (em destaque, sublinhado, na Tabela 2), para 32 clientes enviando arquivos de 100 MB. Justifica-se esta

escolha devido ao aumento na porcentagem de utilização do disco rígido e porcentagem de utilização de memória neste servidor.

Tabela 6 – Informações de sobrecarga do servidor primário

Número de Clientes	Tamanho do Arquivo	Tipo de Operação	Uso inicial do disco (%) P1, P2, P3	Uso final do disco (%) P1, P2, P3	Status
16	50 MB	Escrita	26, 43, 28	35, 52, 37	Normal
16	100 MB	Escrita	26, 43, 28	44, 59, 45	Normal
32	50 MB	Escrita	26, 43, 28	43, 59, 45	Normal
32	100 MB	Escrita	26, 43, 28	59, <u>76</u> , 60	Sobrecarga

P1 – servidor primário 1; P2 – servidor primário 2; P3 – servidor primário 3

Quando a sobrecarga do servidor primário é detectada, um servidor secundário deve ser escolhido com base na fórmula apresentada na Figura 17. Neste teste, os servidores secundários utilizados foram o S1, S2 e S3, apresentados na Tabela 7.

Tabela 7 – Servidores secundários no teste de sobrecarga dos servidores primários

Servidor secundário	Disco utilizado (%)
S1	57%
S2	23%
S3	29%
S4	28%

Desta forma, o servidor secundário escolhido foi o S2 pelo fato de todos os servidores estarem em igualdade em relação ao tráfego de rede, e deste servidor ter mais espaço disponível no disco no ato da solicitação das métricas (23%). Após a escolha, este servidor recebe mensagem para iniciar seu Coletor – servidor primário.

O processo é finalizado com o servidor secundário escolhido solicitando os *metadados* a um servidor primário ativo, escolhido de forma aleatória.

5.6 Validação da avaliação da sobrecarga do servidor secundário

O cenário de validação da sobrecarga do servidor secundário foi composto de três servidores primários, dois servidores secundários e 32 clientes. Os servidores primários, os clientes e um dos servidores secundários têm características iguais às apresentadas na seção 5.2; este servidor secundário é denominado S1. O outro servidor secundário, denominado de S2, foi virtualizado em uma estação com característica semelhante a um de cliente, no entanto com Memória RAM de 256 MB.

O motivo de se manter somente dois servidores secundários neste teste é o fato de que o servidor primário, na operação de escrita de arquivos, escolhe dois servidores secundários para envio das porções que são recebidas. Ao se manter dois servidores secundários, os dois deverão ser escolhidos para o envio das porções e, conseqüentemente, a sobrecarga será avaliada.

Sobre este cenário foi simulada a operação de escrita de arquivos (pelo fato de alterar a ocupação e a atividade em disco; que fazem parte da métrica para a definição da sobrecarga do servidor secundário) com o tamanho de 50 MB e 100 MB através de 16 e 32 clientes, com uma frequência de vinte vezes para cada operação, no sentido de verificar a sobrecarga nos servidores secundários.

Nestes testes, o sistema somente foi caracterizado como sobrecarregado no servidor secundário S2 (em destaque, sublinhado, na Tabela 8), para 32 clientes enviando arquivos de 100 MB. Este comportamento se justifica devido ao aumento na porcentagem de utilização do disco rígido e pelo fato de que, nas aferições, ter sido constatado o aumento consecutivo de utilização em disco. Na Tabela 8 são apresentadas informações de sobrecarga.

Tabela 8 – Informações de sobrecarga do servidor secundário

Número de Clientes	Tamanho do Arquivo	Tipo de Operação	Uso inicial do disco (%) S1, S2	Uso final do disco (%) S1, S2	Status
16	50 MB	Escrita	26, 43	35, 52	Normal
16	100 MB	Escrita	26, 43	44, 59	Normal
32	50 MB	Escrita	26, 43	43, 59	Normal
32	100 MB	Escrita	26, 43	59, <u>76</u>	Sobrecarga

S1 – servidor secundário 1; S2 – servidor secundário 2

Nesta validação, os clientes possuem a porcentagem disponível em disco como apresentada na Tabela 9.

Tabela 9 – Clientes na sobrecarga do servidor secundário

Cliente	% Disco	Cliente	% Disco	Cliente	% Disco	Cliente	% Disco
C1	25	C2	15	C3	45	C4	26
C5	30	C6	21	C7	41	C8	56
C9	32	C10	85	C11	26	C12	14
C13	31	C14	29	C15	65	C16	36
C17	26	C18	31	C19	15	C20	40
C21	30	C22	25	C23	17	C24	39
C25	29	C26	17	C27	26	C28	35
C29	36	C30	18	C31	31	C32	36

O cliente escolhido foi o C12 pelo fato de apresentar mais espaço em disco disponível (14%) e ter sido classificado perante a equação apresentada na Figura 17.

O cliente escolhido recebe uma mensagem para iniciar seu Coletor - servidor secundário, fazendo a sincronização dos *metadados* e iniciando seu coletor.

5.7 Considerações finais

Neste capítulo foi apresentada a validação do sistema FlexA desenvolvido após as modificações propostas neste trabalho. Após a validação constatou-se que o sistema cumpre todos os requisitos propostos no que diz respeito à nova forma de escrita e leitura de arquivos, à avaliação da sobrecarga de servidores primários e servidores secundários e a autoavaliação de servidores primários.

No Capítulo 6 que segue, são realizados testes de desempenho nos sistemas de arquivos distribuídos FlexA original, FlexA desenvolvido, Tahoe-LAFS e NFS no sentido de comparar tais sistemas quanto ao desempenho na escrita e leitura de arquivos.

6 AVALIAÇÃO E COMPARAÇÃO DE DESEMPENHO

6.1 Considerações iniciais

O processo de avaliação e comparação de desempenho tem como principal objetivo inserir o FlexA desenvolvido em um cenário para que seu comportamento possa ser comparado com outros sistemas de arquivos distribuídos.

Neste capítulo, inicialmente é apresentado o cenário de realização das avaliações e as avaliações de desempenho dos sistemas de arquivos distribuídos Tahoe-LAFS, NFS, FlexA original e FlexA desenvolvido.

Por fim, são apresentadas as comparações das avaliações realizadas com os diferentes sistemas de arquivos distribuídos. Para a realização da avaliação, optou-se pela coleta de dados para verificação do tempo gasto na distribuição dos arquivos.

6.2 Cenário de avaliação

A avaliação foi realizada utilizando o *cluster* instalado no laboratório do GSPD, que é composto por computadores com diferentes configurações, sendo estes computadores divididos em dois grupos, mencionados na seção 5.2.

O grupo A, com melhor *hardware*, recebeu os clientes e o grupo B ficou responsável por receber os servidores. Cada computador do grupo A recebeu duas máquinas virtualizadas operando, assim, com dois clientes cada uma.

As avaliações de desempenho foram realizadas com os sistemas de arquivos distribuídos Tahoe-LAFS, NFS, FlexA original e FlexA desenvolvido.

A escolha do Tahoe-LAFS tem como motivação o fato deste sistema ter servido como base na concepção do FlexA original; no caso do NFS, pela disponibilidade de código, por apresentar uma arquitetura cliente-servidor convencional e por apresentar arquitetura

semelhante ao FlexA original (sem uso de replicação e com a ausência de verificação de falha do servidor e de sobrecarga).

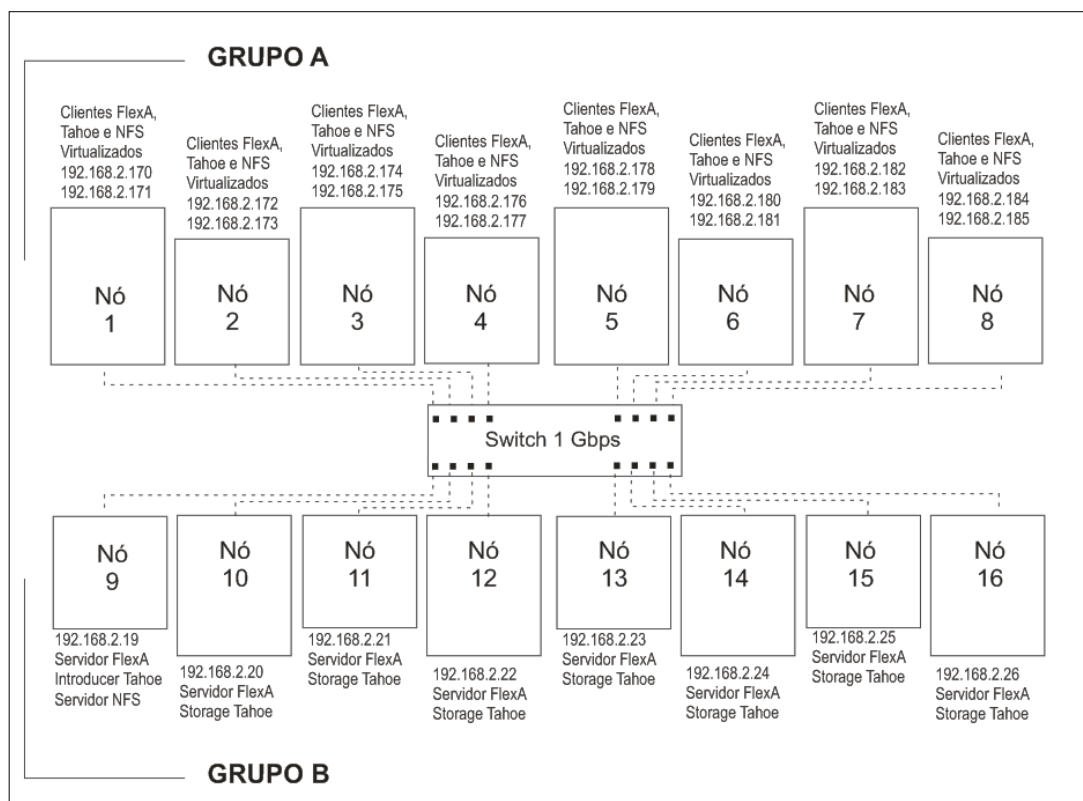
Devido ao fato de que cada sistema de arquivos distribuído necessita de configurações diferenciadas, os servidores de armazenamento foram disponibilizados de formas diferentes para cada sistema avaliado. No Quadro 3 é apresentada a configuração da arquitetura dos sistemas testados.

Quadro 3 – Disposição dos servidores na avaliação

Sistema de Arquivo Distribuído	Disposição dos servidores
FlexA original	3 servidores primários
FlexA desenvolvido	3 servidores primários e 5 servidores secundários
Tahoe-LAFS	1 <i>Introducer</i> e 7 servidores de armazenamento
NFS	1 servidor de armazenamento

A configuração dos servidores e clientes é igual ao cenário de testes apresentado na seção 5.2. A Figura 43 apresenta cenário de avaliação com diferentes disposições, dependendo do sistema de arquivos distribuído a ser testado.

Figura 43 – Cenário de avaliação



Para o FlexA original foram utilizados três servidores primários devido o fato de que essa é a quantidade mínima para garantir a disponibilidade do sistema.

O mesmo acontece para a quantidade de servidores primários constantes no ambiente de testes do FlexA desenvolvido; no entanto, a quantidade de 5 servidores secundários se dá pelo fato de que no mínimo dois servidores são necessários para a replicação de porções e outros três podem ser utilizados na sobrecarga de primários e secundários ou na indisponibilidade de um servidor primário.

Por padrão, o Tahoe-LAFS trabalha com dez estações do tipo servidor de armazenamento (podendo ser modificado); como o *cluster* é composto de 8 estações para este fim, determinou-se 1 estação do tipo *Introducer* e as demais como servidores de armazenamento.

Por fim, no caso do NFS, somente um servidor é necessário para realizar o armazenamento e disponibilização de arquivos aos clientes.

6.3 Avaliação de desempenho

A avaliação ocorreu pela execução das operações de escrita e leitura de arquivos com tamanhos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB, coletando o tempo de cada operação.

As avaliações foram aplicadas e analisadas com a utilização de *script* escrito em linguagem *Python*, responsável pela execução repetitiva das operações de escrita e leitura para os seis tipos de arquivos citados.

A avaliação foi aplicada por meio do envio simultâneo de arquivos pelos clientes envolvidos.

No estudo desenvolvido por Segura (2013) para determinar o número de operações necessárias para a obtenção de resultados estatisticamente confiáveis para este teste, chegou-se ao valor de vinte interações (teste de convergência).

Os resultados foram organizados mostrando, inicialmente, as avaliações realizadas com o FlexA original (sem as modificações propostas), seguido das avaliações realizadas com o Tahoe-LAFS, NFS e finalizando com o FlexA desenvolvido neste trabalho.

6.3.1 FlexA original

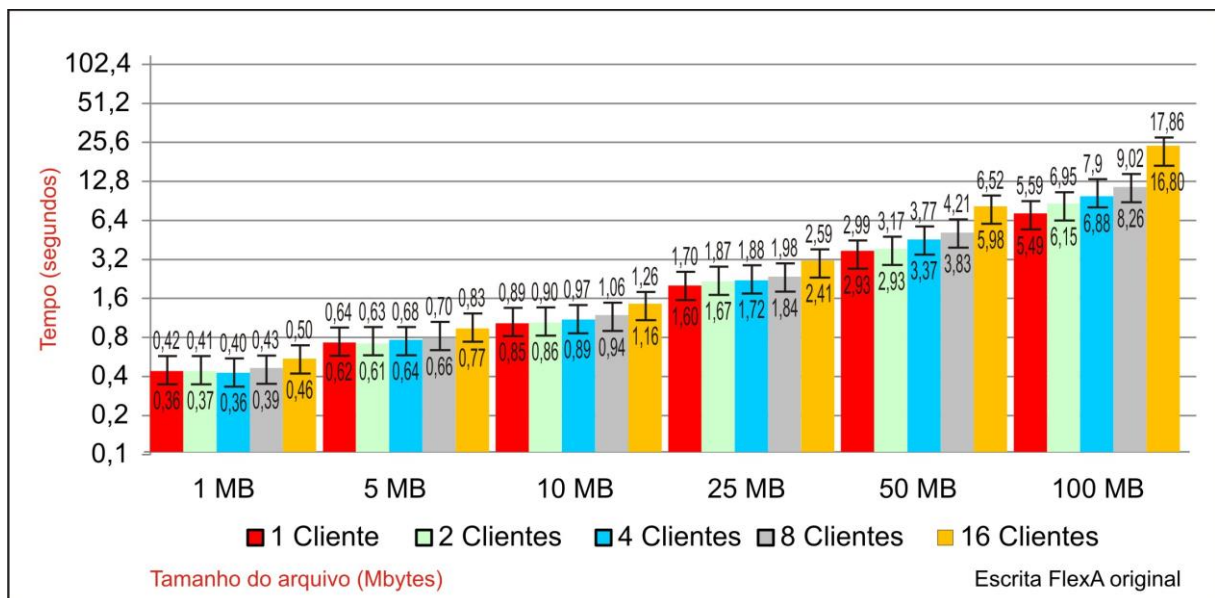
A principal motivação para a realização de testes no FlexA original é a de comparar o tempo das operações de escrita e leitura, mediante as modificações que foram incorporadas no FlexA desenvolvido.

O FlexA original foi avaliado, aferindo-se os tempos nas operações de escrita e leitura para arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB sendo enviados a partir de 1, 2, 4, 8 e 16 clientes.

No Apêndice B são apresentas tabelas com informações obtidas nesta avaliação, incluindo os intervalos com 95% de confiança.

No Gráfico 5 são apresentados tempos de escrita no FlexA original.

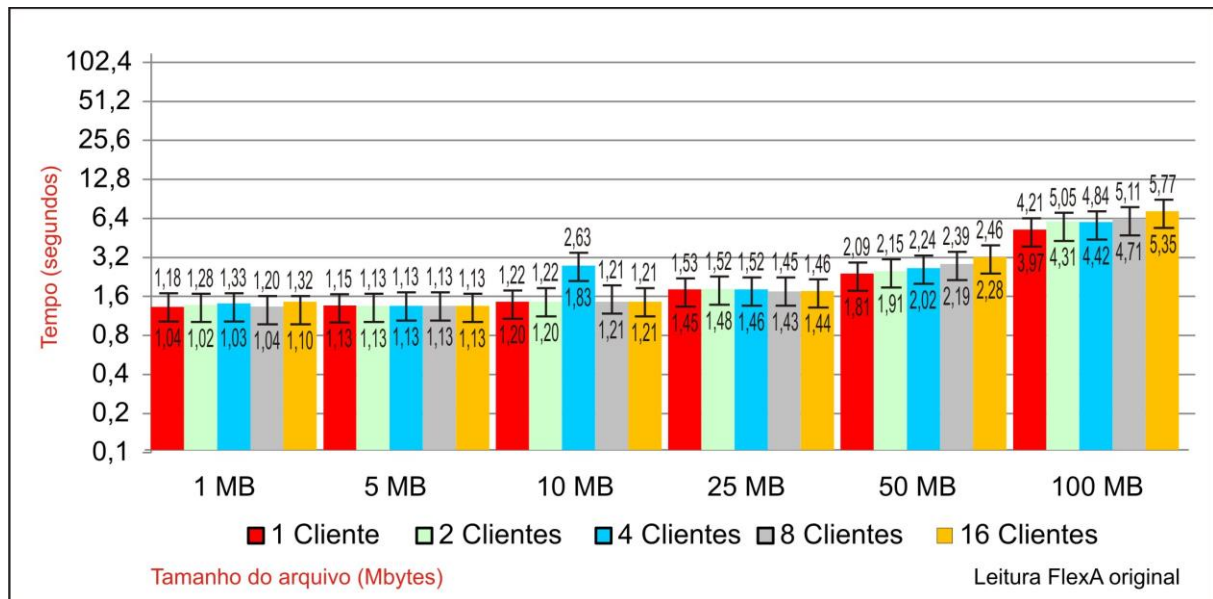
Gráfico 5 – Tempos de escrita no FlexA original



Os tempos de escrita ficam maiores conforme o tamanho dos arquivos aumenta, o que é considerado normal uma vez que arquivos maiores demandam um tempo maior no processo de criptografia, divisão e envio de porções do arquivo pelos clientes.

No Gráfico 6 são apresentados os tempos de leitura para todos os tamanhos de arquivos testados no FlexA original.

Gráfico 6 – Tempos de leitura no FlexA original



De uma forma geral, os tempos são proporcionais ao tamanho do arquivo e ao número de clientes, valores estes, assumidos dentro do intervalo de confiança.

6.3.2 Tahoe-LAFS

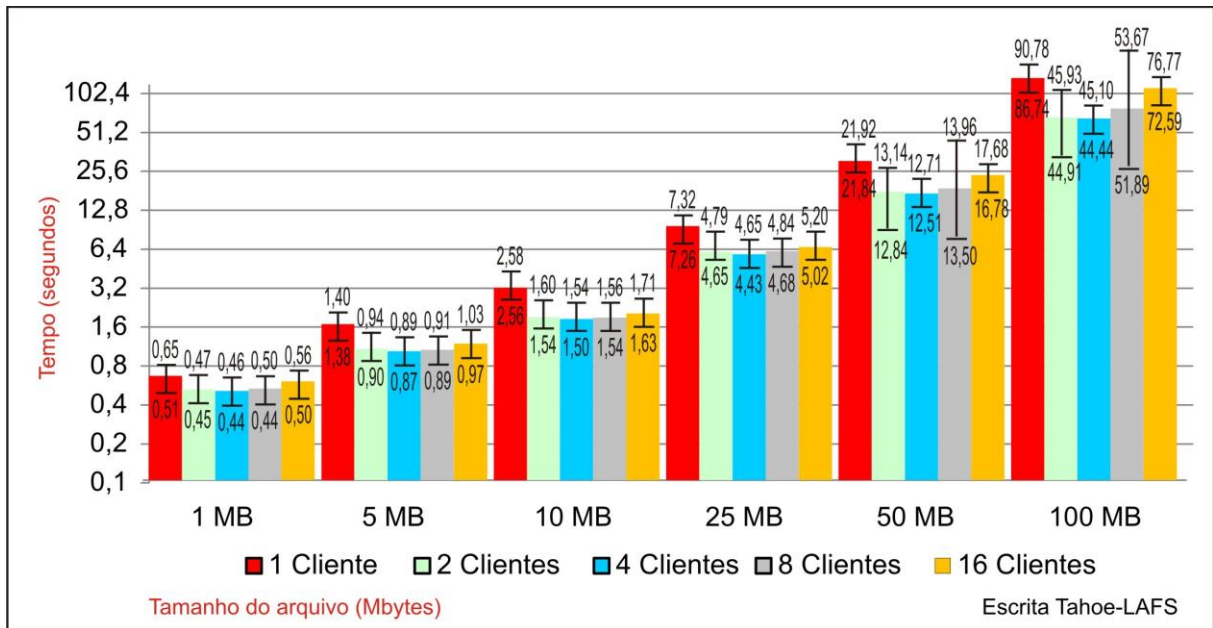
O sistema de arquivos distribuído Tahoe-LAFS foi submetido aos mesmos testes, realizando operações de escrita e leitura de arquivos variando de 1 MB a 100MB com 1, 2, 4, 8 e 16 clientes. No Apêndice B são apresentadas as tabelas com informações obtidas nesta avaliação, incluindo os intervalos com 95% de confiança. No Gráfico 7 são apresentados tempos de escrita no Tahoe-LAFS.

Analisando o gráfico apresentado, nota-se que o tempo na operação de escrita aumenta proporcionalmente quando existe o aumento do número de clientes e o tamanho do arquivo.

Na análise individual do gráfico, verificando o envio de arquivos por 1 cliente, nota-se que o tempo de envio de um arquivo de 1 MB por um cliente é semelhante ao tempo de envio de um arquivo de 1 MB por 16 clientes, considerando o intervalo de confiança.

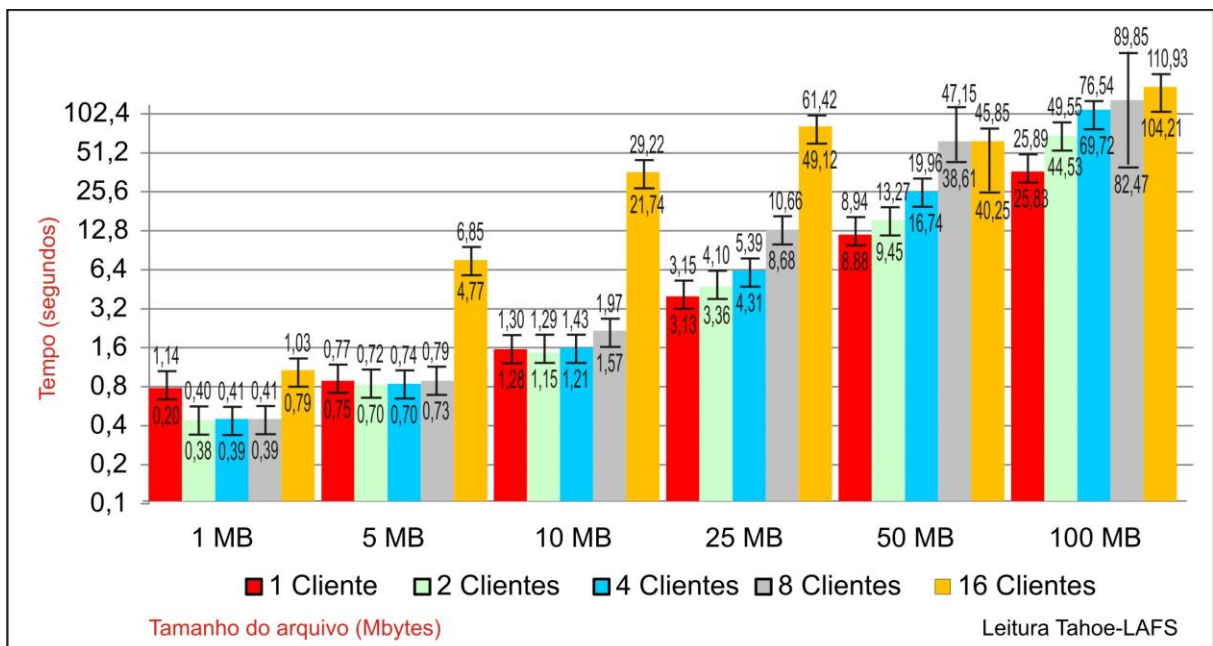
No entanto, para arquivos acima de 5 MB, nota-se que o tempo no envio a partir de 1 cliente é maior, se comparado a 2, 4, 8 e 16 clientes. O mesmo acontece para os demais tamanhos de arquivo.

Gráfico 7 – Tempos de escrita no Tahoe-LAFS



No Gráfico 8 são apresentados tempos de leitura no Tahoe-LAFS para todos os tamanhos de arquivo da avaliação.

Gráfico 8 – Tempos de leitura no Tahoe-LAFS



Na análise geral na operação de leitura, os tempos aumentam à medida que aumenta a quantidade de clientes e o tamanho do arquivo de forma proporcional (ao aumento do número de clientes e tamanho do arquivo). Uma vez analisado individualmente, a leitura de um

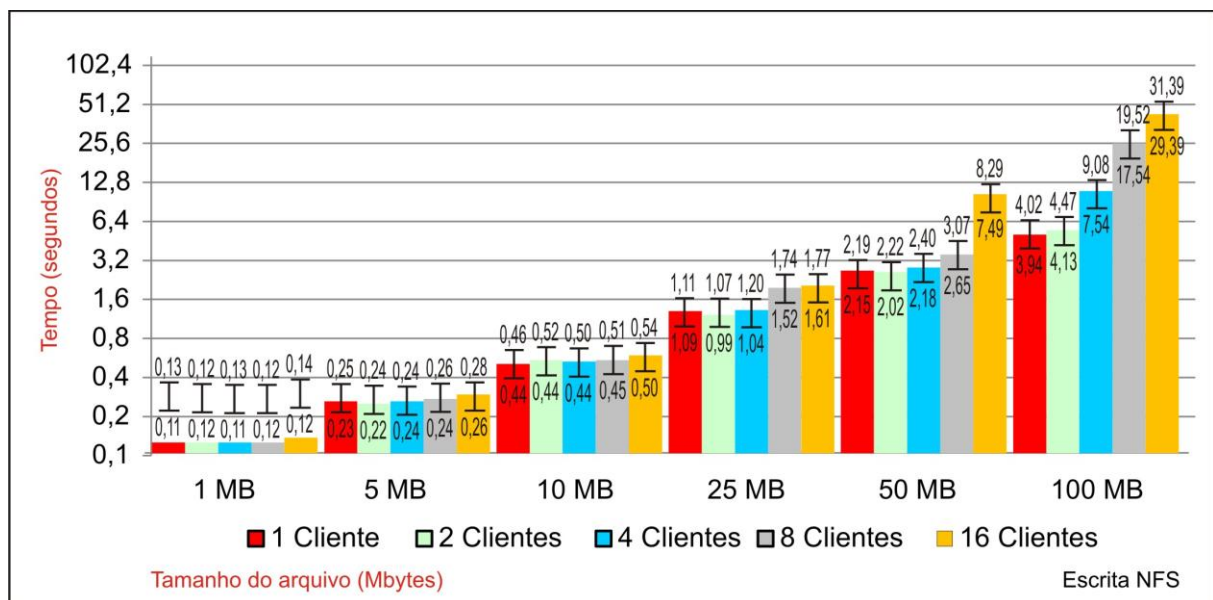
arquivo de 1 MB inicia-se com 0,67 segundos, diminui o tempo até 8 clientes, voltando a aumentar com 16 clientes: o mesmo acontece para arquivos de 5 MB. Para arquivos de 10 MB, o tempo cai para 2 clientes e volta a aumentar a partir de 4 clientes; a partir de 25 MB, o tempo aumenta proporcionalmente ao tamanho do arquivo. Desta maneira, conclui-se que o aumento do tempo está relacionado ao aumento dos arquivos e aumento no número de clientes.

6.3.3 NFS - *Network File System*

O sistema NFS foi submetido ao mesmo tipo de avaliação nas operações de escrita e leitura para arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB por 1, 2, 4, 8 e 16 clientes.

No Apêndice B são apresentadas as tabelas com informações obtidas nesta avaliação, incluindo os intervalos com 95% de confiança. No Gráfico 9 são apresentados tempos de escrita para o sistema de arquivos distribuído NFS.

Gráfico 9 – Tempos de escrita no NFS



No processo de escrita, pelo fato deste sistema de arquivos distribuído centralizar em um servidor as operações, é de se esperar que o tempo aumente à medida que o tamanho dos arquivos e a quantidade de clientes aumentem quando efetuam a escrita de forma simultânea. Quando a análise é realizada individualmente, alguns cenários merecem destaque.

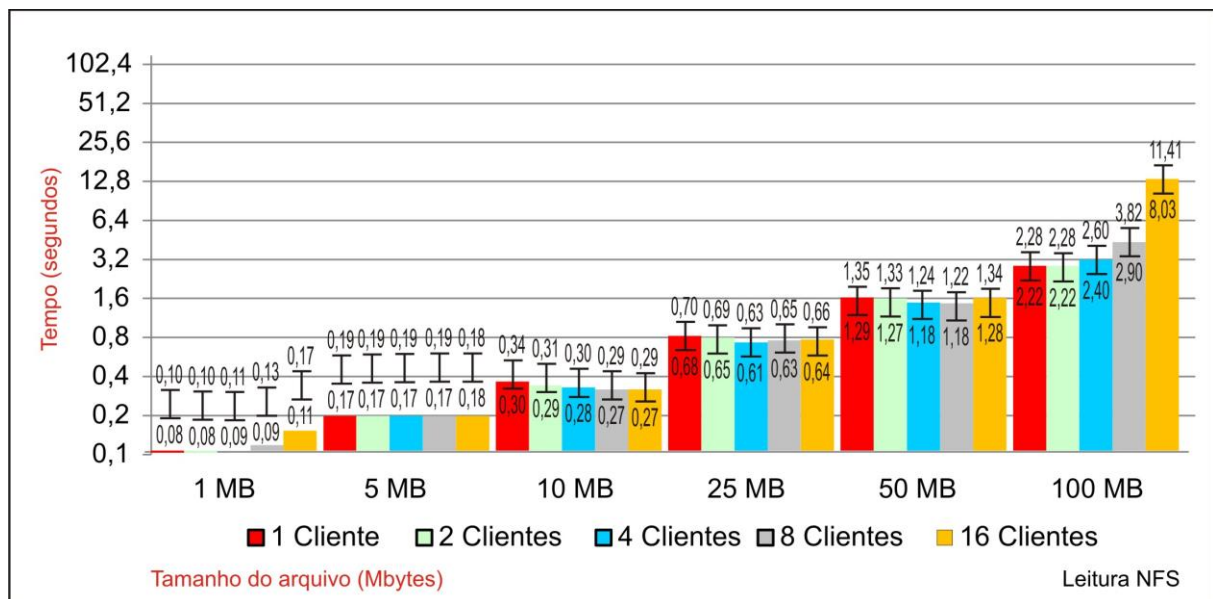
Vejam, como exemplo, o caso de envio de um arquivo de 50 MB a partir de 2 clientes: o tempo é de 2,12 segundos; aumentando para 100 MB, tem-se o tempo de 4,30 segundos: tempo um pouco maior que o dobro, para o dobro do tamanho de um arquivo.

Fazendo-se a mesma análise para 4 clientes, tem-se 2,29 segundos para arquivo de 50 MB e 8,31 segundos para arquivos de 100 MB; para 8 clientes, tem-se 2,86 segundos para 50 MB e 18,53 segundos para 100 MB – um aumento significativo. No caso de 16 clientes, tem-se 7,89 segundos e 30,39 segundos para arquivos de 50 MB e 100 MB, respectivamente.

Com base nestes cenários apresentados, o sistema confirma o que foi previsto: um aumento no tempo conforme aumenta o tamanho do arquivo e o número de clientes o que, em alguns casos, mostra um aumento exagerado no tempo; reflexo da centralização dos atendimentos no servidor.

Complementando a descrição dos testes do sistema de arquivos distribuído NFS, no Gráfico 10 são apresentados tempos de leitura para arquivos que vão de 1 MB a 100 MB com 1, 2, 4, 8 e 16 clientes.

Gráfico 10 – Tempos de leitura no NFS



A operação de leitura tem tempos menores se comparados à operação de escrita. No entanto, a proporcionalidade de tempo discutida na operação de escrita, se mantém, ou seja, pelo fato deste sistema de arquivos distribuído centralizar em um servidor as operações, o tempo aumenta de forma proporcional quando olha-se para os arquivos de forma coletiva, tempo esse que se torna, às vezes desproporcional, quando analisado cada tipo de arquivo individualmente. Como exemplo, cita-se o tempo de 1,31 segundos para envio de um arquivo

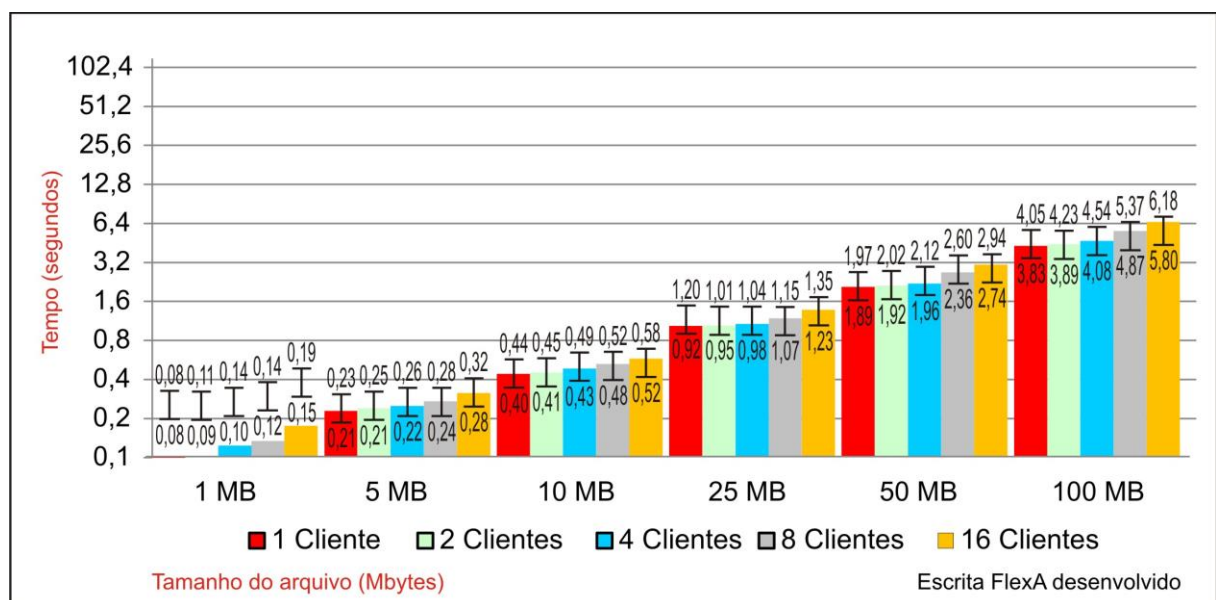
com tamanho 50 MB a partir de 16 clientes e 9,72 para arquivos com 100 MB e mesmo número de clientes; na leitura de um arquivo de 50 MB, para 1 cliente tem-se um tempo de 1,32 segundos e 1,31 segundos para 16 clientes, ou seja, mesmo aumentando o número de clientes, o tempo continuou semelhante. Por fim, para confirmar a desproporcionalidade, para leitura de um arquivo de 5 MB, o tempo se mantém em 0,18 segundos para todas as quantidades de cliente.

6.3.4 FlexA desenvolvido

O sistema FlexA desenvolvido foi submetido às mesmas avaliações mencionadas anteriormente para os sistemas FlexA original, Tahoe-LAFS e NFS. No Apêndice B são apresentadas as tabelas com informações obtidas nesta avaliação, incluindo os intervalos com 95% de confiança.

No Gráfico 11 são apresentados tempos no processo de escrita no sistema FlexA desenvolvido.

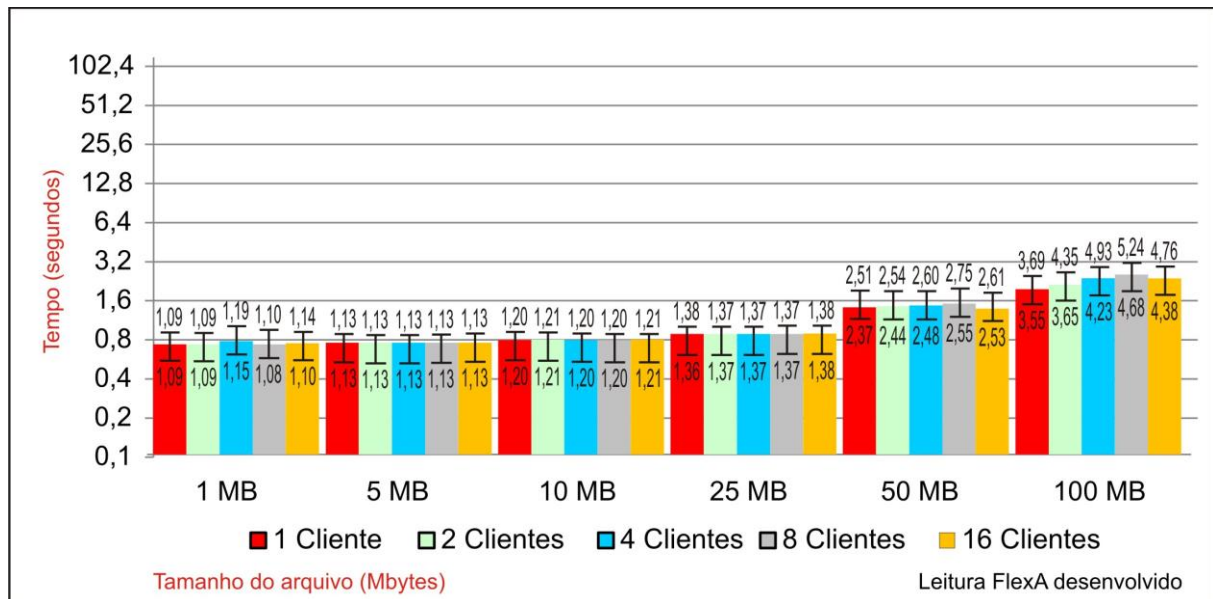
Gráfico 11 – Tempos de escrita no FlexA desenvolvido



No processo de escrita, os tempos aumentam à medida que os arquivos tornam-se maiores e o aumento do número de clientes aumenta. Como exemplo, cita-se o envio a partir de 8 clientes, com tempo de 2,48 segundos para arquivo de 50 MB e 5,12 segundos para arquivo de 100 MB. Com 16 clientes, tem-se 2,84 segundos e 5,99 segundos para arquivos de

50 MB e 100 MB, respectivamente. O aumento proporcional se deve ao fato de que o tempo para realizar a criptografia, dividir e enviar as porções aos servidores aumenta conforme aumenta o tamanho do arquivo, alvo deste processo. No Gráfico 12 são apresentados tempos de leitura para o FlexA desenvolvido.

Gráfico 12 – Tempos de leitura no FlexA desenvolvido



De uma forma geral, a operação de leitura tem tempos maiores conforme aumenta a quantidade de clientes e o tamanho do arquivo; ou seja, o aumento do tempo é proporcional ao tamanho do arquivo e ao número de clientes.

No entanto, quando a comparação é feita visualizando os tamanhos dos arquivos de forma individual, observam-se alguns aspectos: na leitura de 1 MB, 5 MB, 10 MB e 25 MB, para qualquer quantidade de clientes, o tempo mantém uma variação mínima de acréscimo ou decréscimo.

Por exemplo, na leitura de arquivos de 5 MB, o tempo se mantém em 1,13 segundos; na leitura de 25 MB, varia entre 1,37 segundos e 1,38 segundos. No entanto, para arquivos de 50 MB e 100 MB, observa-se um aumento. Para arquivos de 100 MB para um cliente, 3,62 segundos e 4,57 para 16 clientes.

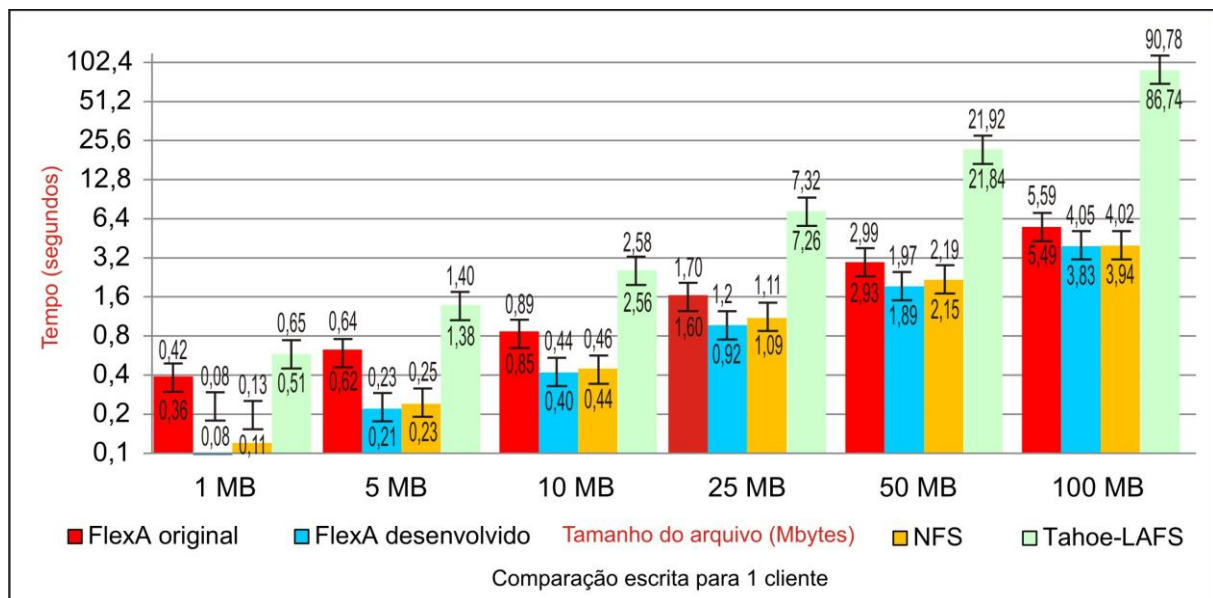
Pelo fato da variação do tempo não ser consideravelmente grande, a leitura é vista como eficiente.

6.4 Comparação de desempenho

Após a fase de avaliação dos sistemas de arquivos distribuídos abordados na seção anterior, nesta seção são apresentados os comparativos entre os sistemas, juntamente com as devidas considerações. Inicialmente é apresentada comparação para envio de arquivos a partir de um cliente, sendo que no Apêndice B são apresentadas as tabelas com informações desta comparação, incluindo os intervalos com 95% de confiança.

Para uma melhor visualização, no Gráfico 13 é apresentada a comparação dos tempos na operação de escrita de 1 cliente para os sistemas de arquivos distribuído avaliados.

Gráfico 13 – Comparação do tempo de escrita para 1 cliente



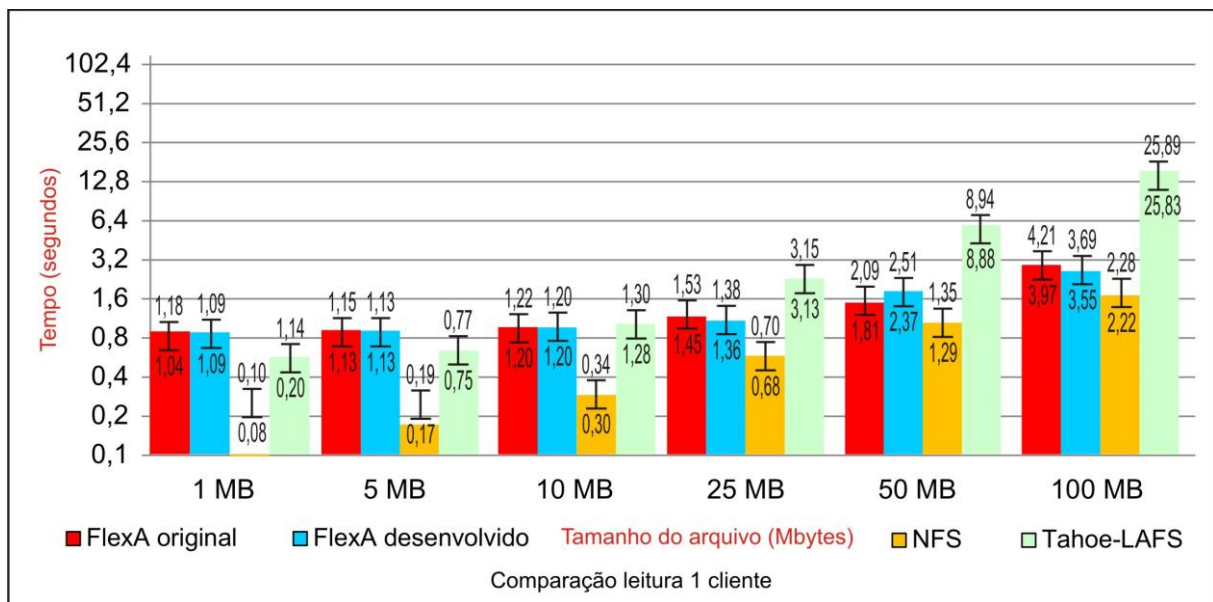
No processo de escrita para 1 cliente, o FlexA desenvolvido apresentou os melhores resultados para todos os tamanhos de arquivos testados, mesmo oferecendo replicação de porções, avaliação do servidor primário e avaliação das sobrecargas.

O NFS, mesmo não replicando arquivos e não realizando operações para garantir a disponibilidade do servidor, é classificado em segundo lugar.

O FlexA original foi classificado em terceiro lugar, mesmo oferecendo igual processo de criptografia do FlexA desenvolvido e não oferecendo mecanismos de replicação e garantia de disponibilidade dos servidores; dessa forma, apresentou-se com tempos de envio maiores que o FlexA desenvolvido em todos os cenários de testes.

No caso do Tahoe-LAFS, mesmo apresentando características semelhantes ao FlexA (nas duas versões), pelo fato de dividir o arquivo e enviar porções aos servidores de armazenamento, apresentou-se mais lento que os demais. No entanto, as versões do FlexA não tem o papel de um centralizador, como o Introducer no Tahoe-LAFS. No Gráfico 14 é apresentada a comparação do tempo de leitura para 1 cliente.

Gráfico 14 – Comparação do tempo de leitura para 1 cliente



Na leitura de arquivos, o NFS foi o mais rápido se comparado ao FlexA original, FlexA desenvolvido e Tahoe-LAFS. Pelo fato de não realizar criptografia dos arquivos e por centralizar as operações somente em um servidor.

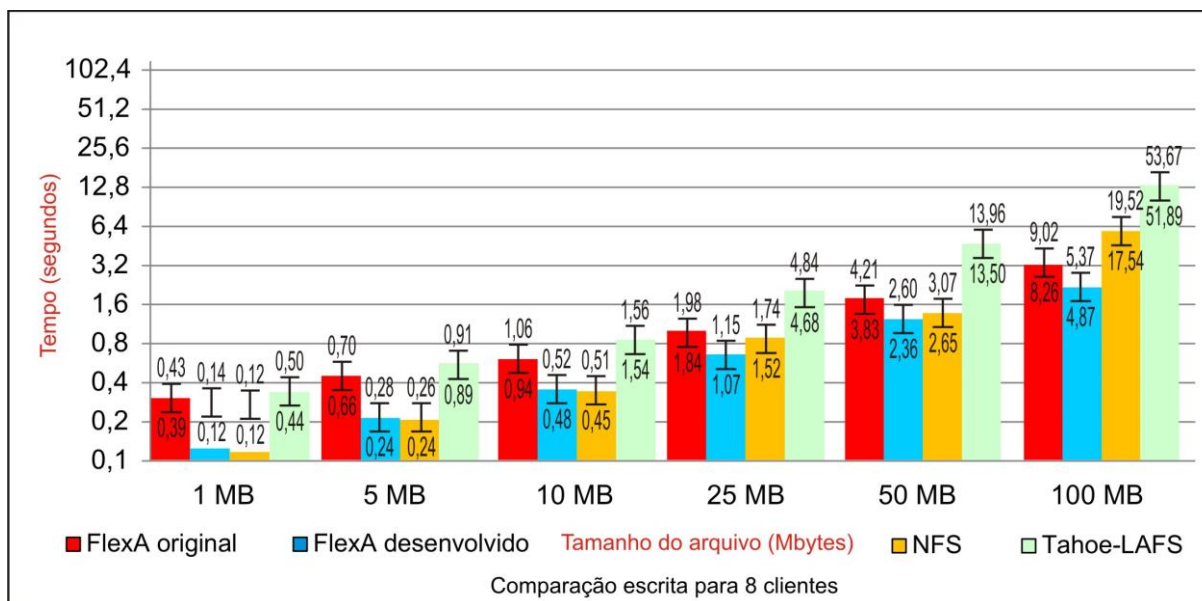
O FlexA desenvolvido ficou em segundo lugar mesmo fazendo uso de criptografia, classificação dos melhores servidores para solicitar as porções do arquivo e utilizar o Grupo de Réplicas.

O próximo colocado foi o FlexA original, pelo fato de ser necessária a leitura de porções dos arquivos em dois servidores primários de forma concorrente para solicitação dos arquivos e posterior união e descriptografia do arquivo antes de disponibilizar ao usuário.

O Tahoe-LAFS apresentou-se o mais lento, mesmo trabalhando de forma semelhante às duas versões do FlexA. Destaca-se aqui tempos altos se comparados aos do FlexA desenvolvido: no caso do envio de arquivos de 100 MB a partir de 1 cliente, o FlexA desenvolvido apresentou 3,62 segundos e o Tahoe-LAFS 25,86 segundos.

Continuando a apresentar a comparação entre os sistemas, tem-se os tempos das operações de escrita e leitura com 8 clientes. As tabelas comparativas são apresentadas no Apêndice B. No Gráfico 15 é apresentada comparação entre os sistemas, com base no tempo de escrita para 8 clientes.

Gráfico 15 – Comparação do tempo na escrita de 8 clientes



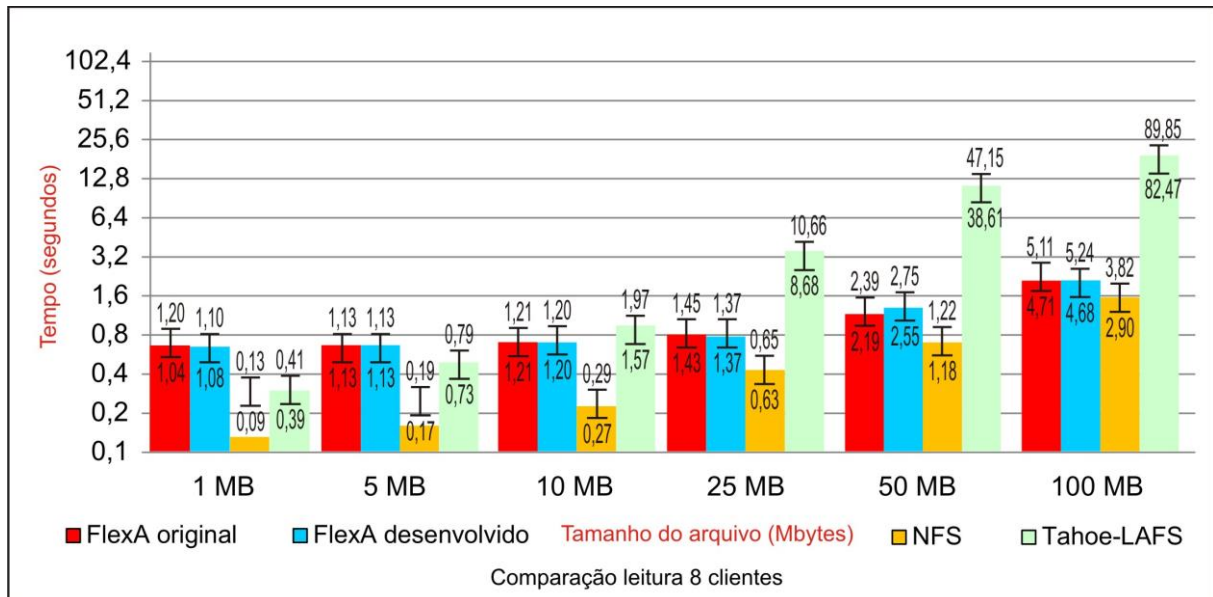
No cenário para a operação de escrita com 8 clientes para arquivos de 1 MB, 5 MB e 10 MB, o NFS e FlexA desenvolvido apresentaram tempos equivalentes. No entanto, a partir de arquivos com 25 MB, o FlexA desenvolvido é o mais rápido. Destaca-se o envio de arquivos de 100 MB: com 5,2 segundos para o FlexA desenvolvido e 18,53 segundos para o NFS. Isto se dá pelo fato de que, aumentando o número de clientes, toda a carga de trabalho centraliza-se em um servidor no NFS, diferentemente do FlexA desenvolvido, que utiliza mais servidores na escrita de arquivos.

O FlexA original aparece em seguida. Quando comparado ao NFS, este sistema é mais lento devido ao uso de criptografia, divisão dos arquivos e envio das porções. Quando comparado com o FlexA desenvolvido, o FlexA original se mostrou mais lento mesmo não tratando a questão da avaliação de servidores e sobrecarga. No entanto, o processo de disponibilidade se distingue para ambas as versões.

O Tahoe-LAFS foi o mais lento neste cenário, quando comparado ao FlexA desenvolvido, é dez vezes mais lento para arquivos de 100 MB: 5,12 segundos no FlexA desenvolvido e 52,78 segundos no Tahoe-LAFS.

No Gráfico 16 é apresentada comparação de tempo na leitura para 8 clientes.

Gráfico 16 – Comparação do tempo na leitura de 8 clientes



Na operação de leitura, o NFS foi o sistema mais rápido para todos os tamanhos de arquivo pelo fato de apresentar arquitetura mais simples e não dispor de mecanismo de disponibilidade e serviços de verificação do servidor.

Em seguida, o Tahoe-LAFS apresentou tempos melhores para arquivos de 1 MB e 5 MB. No entanto, a partir de arquivos de 10 MB até 100 MB, o FlexA desenvolvido foi melhor classificado. O FlexA original vem classificado a seguir e, por último, para arquivos de 10 MB, 25 MB, 50 MB e 100 MB, o Tahoe-LAFS é classificado como o mais lento.

Nota-se que, conforme o número de clientes e o tamanho do arquivo aumentam, a proporcionalidade do FlexA desenvolvido se mantém, ao contrário do Tahoe-LAFS.

Como exemplo, tem-se a leitura de um arquivo de 25 MB com tempo de 9,67 segundos e de 42,88 segundos para um arquivo de 50 MB no Tahoe-LAFS. No FlexA desenvolvido, tem-se 1,37 segundos para um arquivo de 25 MB e 2,65 segundos para um arquivo de 50 MB.

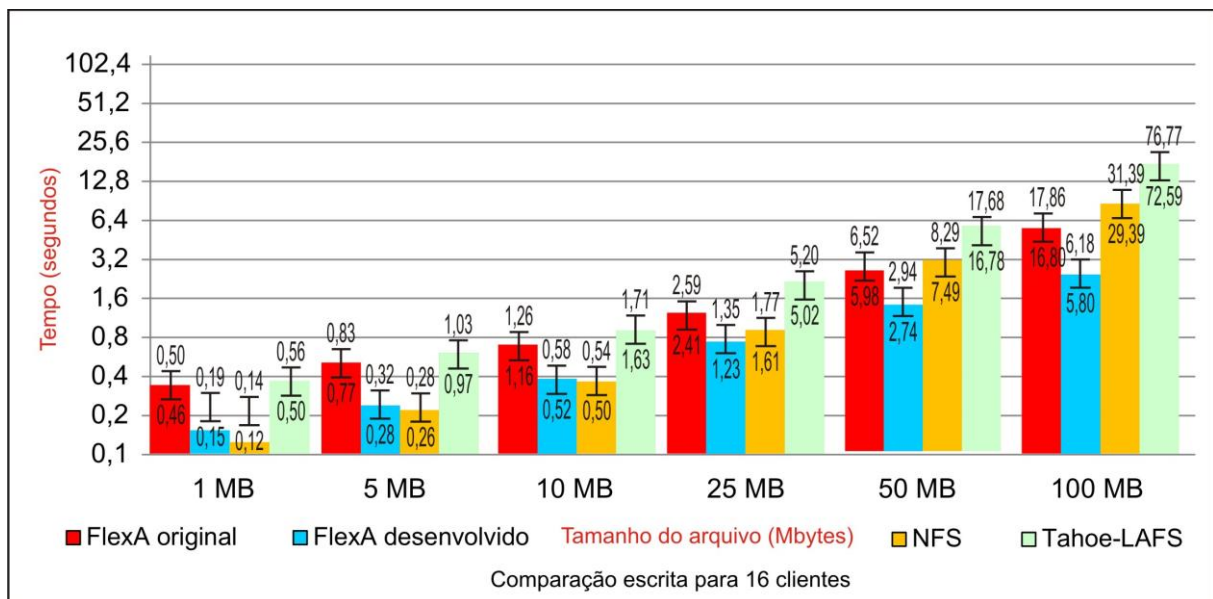
Por fim, apresentam-se as comparações nas operações de escrita e leitura para 16 clientes com arquivos de 1 MB a 100 MB. A Tabela contendo os tempos é apresentada no Apêndice B. No Gráfico 17 é apresentado o cenário no processo de escrita.

Aumentando o número de clientes para 16, verifica-se que o Tahoe-LAFS volta a apresentar o pior desempenho. O caso mais custoso é o envio de arquivos com 100 MB, com 74,68 segundos, contra 5,99 segundos do FlexA desenvolvido.

Os sistemas que apresentaram tempos melhores foram o NFS e o FlexA desenvolvido. O NFS se mantém mais rápido para arquivos até 10 MB. No entanto, para arquivos a partir de 25 MB, o FlexA desenvolvido foi o mais rápido. Isto é explicado pelo fato de que o NFS tem o sistema sobrecarregado pelo envio a partir de 16 clientes, tendo as requisições centralizadas em um único servidor; diferentemente do FlexA desenvolvido que, apesar de apresentar criptografia e uso de mecanismos para garantir a disponibilidade, faz a divisão da carga entre três servidores primários e secundários; fato este percebido quando o tamanho do arquivo aumenta e o tempo se torna menor que o NFS.

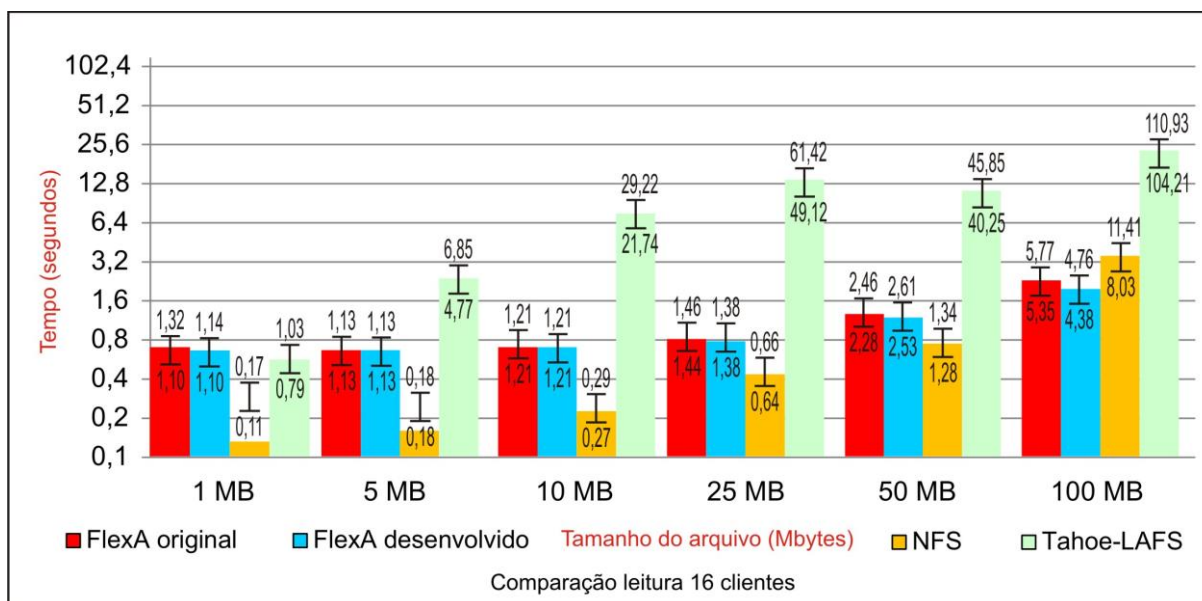
O FlexA original continuou na terceira colocação, melhor posicionado com relação ao Tahoe-LAFS. O Tahoe-LAFS apresentou tempos maiores na escrita de arquivos em todos os casos.

Gráfico 17 – Comparação do tempo na escrita de 16 clientes



O Gráfico 18 apresenta tempos na leitura com 16 clientes. O arquivo com maior tamanho (100 MB) foi o que limitou a divisão da classificação. Até arquivos de 50 MB, a classificação tem o NFS classificado na primeira posição, seguido do FlexA desenvolvido, FlexA original e do Tahoe-LAFS. No entanto, para arquivos de 100 MB, o FlexA desenvolvido foi o mais rápido, seguido do FlexA original, NFS e Tahoe-LAFS. Fica evidente, assim, que o FlexA desenvolvido lida com arquivos maiores e com mais clientes de forma mais eficiente.

Gráfico 18 – Comparação do tempo na leitura de 16 clientes



6.5 Considerações finais

Na análise entre o FlexA original e o FlexA desenvolvido, este último apresentou tempos melhores mesmo com as diversas modificações propostas.

Na operação de escrita, o FlexA original fazia o uso de três servidores primários, cada qual recebendo 2/3 de um arquivo. No FlexA desenvolvido, cada servidor primário passou a receber 1 porção do arquivo, sendo replicadas para dois servidores secundários escolhidos com base em métricas. Cabe ressaltar que o FlexA desenvolvido ainda dispõe de mecanismos para a autoavaliação de servidores primários e avaliação de sobrecarga de servidores primários e secundários. Diante da modificação na forma de armazenar os arquivos e agregação de novas funcionalidades visando melhorar a disponibilidade do sistema, o FlexA desenvolvido apresentou-se mais rápido que o FlexA original em todos os casos comparados.

Na operação de leitura, o FlexA original fazia uso de dois servidores primários, para a leitura de uma porção em um servidor primário e duas porções em outro servidor primário, pelo fato de que as porções são armazenadas na proporção de 2/3 em cada servidor. No FlexA desenvolvido, as três porções passaram a ser lidas através de três servidores diferentes, no entanto, classificados com base no espaço em disco e utilização da rede. Diante dessa nova forma de funcionamento, o FlexA desenvolvido obteve tempos semelhantes quando comparado ao FlexA original, sendo que em alguns momentos foi mais rápido.

Na comparação com o Tahoe-LAFS, o FlexA desenvolvido apresentou melhores tempos nas operações de escrita e leitura na grande maioria das avaliações. Cabe ressaltar que o FlexA desenvolvido faz criptografia dos arquivos, assim como o Tahoe-LAFS e trata a questão da disponibilidade de fato através da replicação de porções.

O principal concorrente foi o NFS, embora o FlexA tenha apresentado tempos melhores em alguns cenários. No entanto, este sistema possui arquitetura mais simples e não oferece a replicação de arquivos, criptografia de arquivos e mecanismos para manter sua disponibilidade.

De uma forma geral, se comparado ao FlexA original, as modificações no FlexA desenvolvido refletem melhoras não somente no tempo nas operações de escrita e leitura, mas também pelo fato de passar a oferecer serviços, antes não oferecidos pelo FlexA original: uso do grupo de Réplicas (passando a tratar a disponibilidade de fato), autoavaliação dos servidores primários e avaliação da sobrecarga dos servidores primários e servidores secundários.

7 CONCLUSÃO

Neste trabalho foram apresentadas a definição e a implantação de melhorias no que diz respeito à questão da disponibilidade no sistema de arquivos distribuído flexível e adaptável FlexA original.

Após estudos realizados no sistema FlexA original, foi detectado que este sistema trata a questão da disponibilidade com a utilização de *cache* e através do modo em que as porções de um arquivo são enviadas aos servidores, sendo 2/3 para cada servidor, após a criptografia e divisão do arquivo. No entanto, não abordava outras questões ligadas à disponibilidade.

Dessa forma, este trabalho concentrou esforços no sentido de oferecer melhorias na questão da disponibilidade para este sistema, culminando na implementação do FlexA desenvolvido. Destaca-se a implementação do Grupo de Réplicas; este especificado no projeto do FlexA original, mas não desenvolvido. Nesta fase, as operações de escrita e leitura sofreram modificações para se adequar à utilização deste grupo, passando a oferecer o balanceamento de carga nos servidores na operação de escrita.

Outros itens agregados foram a autoavaliação dos servidores primários e a avaliação da sobrecarga dos servidores primários e servidores secundários.

Dessa forma, essas funcionalidades abordadas no FlexA desenvolvido fizeram com que este sistema passasse a tratar a questão da disponibilidade de uma melhor forma, sendo o sistema validado nas operações de escrita e leitura, na autoavaliação de servidores primários e na avaliação da sobrecarga de servidores primários e servidores secundários.

Na fase de avaliação, o sistema FlexA desenvolvido foi comparado com outros dois sistemas de arquivos distribuídos, sendo o Tahoe-LAFS (considerado seu precursor) e o outro, o NFS, com o intuito de obter uma comparação com um sistema exclusivamente cliente-servidor. O FlexA desenvolvido mostrou-se mais rápido nas operações de escrita e leitura de arquivos com diversificados tamanhos e número de clientes se comparado ao Tahoe-LAFS. Quando comparado ao seu antecessor (FlexA original), o FlexA desenvolvido apresentou melhores tempos nas operações de leitura e escrita na grande maioria dos testes, mesmo

oferecendo o tratamento da questão da disponibilidade de fato e utilizando mecanismos para a garantia de manutenção dos servidores primários e secundários.

Nenhum dos sistemas de arquivos distribuídos apresentou necessidades especiais no que diz respeito à instalação: por padrão o NFS e Tahoe-LAFS já fazem parte dos repositórios de algumas distribuições Linux. No caso do FlexA, nas duas versões, a instalação compreende a instalação do interpretador *Python*, além de pacotes específicos para administrar a interface de rede, criptografia e obtenção de métricas de servidores e informações de disco, memória e rede dos servidores.

Através das modificações citadas neste trabalho e após as avaliações realizadas, considera-se que o FlexA desenvolvido teve sua questão de disponibilidade melhorada através da efetiva implantação do Grupo de Réplicas, agregação da autoavaliação de servidores primários e avaliação de sobrecarga de servidores primários e servidores secundários.

Visando melhorar o sistema de arquivos distribuído FlexA desenvolvido, abaixo são listados itens que podem ser incorporados ao sistema em trabalhos futuros:

- Na questão da sobrecarga do servidor primário, definir ações a serem tomadas quando o servidor primário estiver perto de sua sobrecarga total, a fim de preservar o servidor ativo;
- Realizar o tratamento para operações desconectadas;
- Definir uma interface gráfica interativa, para a visualização dos componentes do sistema, características dos servidores e manipulação de configurações básicas do sistema tais como número de servidores primários ativos, número de servidores secundários ativos, dentre outros;
- Incorporar algoritmo de criptografia em GPU no FlexA desenvolvido;
- Iniciar estudos para resolver a questão de porções órfãs nos servidores do Grupo de Escrita e do Grupo de Réplicas;
- Comparar o FlexA desenvolvido com outros sistemas de arquivos distribuídos;
- Implementação do cliente FlexA desenvolvido para que o sistema seja utilizado a partir de dispositivos móveis;
- Trabalhar com histórico de versões de arquivos;
- Realizar estudos no sentido de determinar o comportamento do sistema quando servidores primários e servidores secundários deixarem de estar sobrecarregados.

Referências

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 5462**: Confiabilidade e manutenibilidade: terminologia. Rio de Janeiro, 1994.

BAI, Songlin; WU, Hao. The performance study on several distributed file systems. In: INTERNATIONAL CONFERENCE ON CYBER-ENABLED DISTRIBUTED COMPUTING AND KNOWLEDGE DISCOVERY, 2011, Beijing. **Proceedings...** Beijing: IEEE Computer Society, 2011. p. 226-229.

BZOCH, Pavel; SAFARIK, Jiri. Security and reliability of distributed file systems. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT DATA ACQUISITION AND ADVANCED COMPUTING SYSTEMS: technology and applications, 6., 2011a, Prague. **Proceedings...** Prague: IEEE, 2011. p. 764-769.

BZOCH, Pavel; SAFARIK, Jiri. State of the art in distributed file systems: increasing performance. In: EASTERN EUROPEAN REGIONAL CONFERENCE ON THE ENGINEERING OF COMPUTER BASED SYSTEMS, 2., 2011b, Budapest. **Proceedings...** Budapest: IEEE Computer Society; Budapest University of Technology and Economics, 2011. p. 153-154.

CARVALHO, Lúcio Rodrigo; LOBATO, Renata Spolon; MANACERO JUNIOR, Aleardo. Sistema de arquivo flexível e adaptável: um estudo da escalabilidade do sistema FlexA. In: WORKSHOP DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DA UNESP, 3., 2013, Rio Claro. **Tópico temático...** Rio Claro: UNESP, [Universidade Estadual Paulista “Júlio de Mesquita Filho”, Rio Claro] 2013.

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas distribuídos: conceitos e projetos**. 4.ed. Porto Alegre: Bookman, 2007.

FERNANDES, Silas Evandro Nachif. **Sistema de arquivos flexível e adaptável**. 2012. 72f. Dissertação (Mestrado em Ciência da Computação)-Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto, 2012.

FERNANDES, Silas Evandro Nachif et al. A flexible and adaptable distributed file system. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED PROCESSING TECHNIQUES AND APPLICATIONS, 2013, Las Vegas. **Proceedings...** Las Vegas: UCMSS, 2013. p.258-263.

GHEMAWAT, Sanjay; GOBIOFF, Howard; LEUNG, Shun-Tak. The Google file system. In: ACM SYMPOSIUM ON OPERATING SYSTEMS PRINCIPLES, 9., 2003, New York.

Proceedings... New York: ACM New York, 2003. p.29-43.

HARRINGTON, Anthony; JENSEN, Christian. Cryptographic access control in a distributed file system. In: **THE ACM SYMPOSIUM ON ACCESS CONTROL MODELS AND TECHNOLOGIES (SACMAT)**, 8., 2003, Villa Gallia. **Proceedings...** New York. ACM New York, 2003. p. 158-165.

HOWARD, John H. An overview of the Andrew file system. In: USENIX WINTER TECHNICAL CONFERENCE, 1988, Dallas. **Proceedings...** Dallas: USENIX Association, 1988. p. 1-6.

JIAN, Sun; ZHAN-HUAI, Li; XIAO, Zhang. The performance optimization of Lustre file system. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND EDUCATION, 7., 2012, Melbourn. **Proceedings...** Melbourne: IEEE, 2012. p. 214-217.

LOGAN, Jeremy; DICKENS, Phillip. Towards an understanding of the performance of MPI-IO in Lustre file system. In: IEEE INTERNATIONAL CONFERENCE ON CLUSTER COMPUTING, 2008, Maine. **Proceedings...** Maine: IEEE, 2008. p. 330-335.

MOLINA, Hector Garcia. Elections in a distributed computing system. **IEEE Transactions on Computers**, Piscataway, v.31, n.1, p.48-59, Jan. 1982.

MULLER, Gilles et al. Fast, optimized sun RPC using automatic program specialization. **Rapports de Recherche. INRIA**, Le Chesnay, n.3220, juil. 1997. Disponível em: <<http://hal.inria.fr/docs/00/07/34/69/PDF/RR-3220.pdf>>. Acesso em: 01 Outubro 2014.

NEUMAN, B. Clifford; TS'O, Theodore. Kerberos: an authentication service for computer networks. **IEEE Communications Magazine**, Piscataway, v.32, n.9, p.33-38, set. 1994.

OKADA, Thiago Kenji. **Metodologia para recuperação de falhas e garantia de disponibilidade no FlexA**. 2013. 74f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)-Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista "Júlio de Mesquita Filho", São José do Rio Preto, 2013.

ORACLE Corporation. **Lustre File System 1.6 Software**. [Califórnia]: Oracle Corporation, 2014. Disponível em: <<http://docs.oracle.com/cd/E19091-01/lustre.fs16/index.html>>. Acesso em: 01 set. 2014.

OSADZINSKI, Alex. GFS: evolution on Fast-Forward. **Communications of the ACM**, New York, v.53, n.3, p.42-49, mar. 2010.

PUTTER, P.; ROOS, J. D. Relationships: implementing transparency in distributed management systems. In: INTERNATIONAL WORKSHOP ON SYSTEMS MANAGEMENT, 1, 1993, Los Angeles. **Proceedings...** Los Angeles: IBM Press, 1994. p.118-124.

RIVEST, Ronald. The MD5 Message-Digest Algorithm. **Request for Comments (RFC). Informational**, Marina Del Rey, n.1321, Apr. 1992. Disponível em: <<http://www.ietf.org/rfc/rfc1321.txt>>. Acesso em: 23 outubro 2014.

SEGURA, Danilo Costa Marin. **Detecção de falhas de comunicação e balanceamento de carga no FlexA**. 2013. 70f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)-Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto, 2013.

SEGURA, Danilo Costa Marin et al. FlexA: grupo de réplicas em um sistema de arquivos distribuído. In: ESCOLA REGIONAL DE ALTO DESEMPENHO, 4., 2013, São Carlos. **Anais...** São Carlos: UFSCar, 2013. p.143-146.

SHEPLER, S. et al. **Network File System (NFS) version 4 Protocol. Request for Comments (RFC)**, Marina Del Rey, n.3530, Apr. 2003. Disponível em: <<ftp://ftp.rfc-editor.org/in-notes/rfc3530.txt>>. Acesso em: 05 out. 2014. TAHOE-LAFS. Welcome to the Least-Authority File System. Disponível em: <<http://tahoe-lafs.org/>>. Acesso em: 16 jun. 2012.

TANENBAUM, Andrew S.; STEEN, Maarten Van. **Sistemas distribuídos: princípios e paradigmas**. 2.ed. São Paulo: Pearson Prentice Hall, 2007.

TOBBICKE, Rainer. Distributed file systems: focus on Andrew File System/Distributed File Service (AFS/DFS). In: IEEE SYMPOSIUM ON MASS STORAGE SYSTEMS, 13., 1994, Anney. **Proceedings...** Michigan: IEEE Computer Society Press, 1994. p. 23-26.

WHITE, Robert. Fault tolerance in distributed power systems. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 25., 2003, Portland. **Proceedings...** Washington, DC: IEEE Computer Society, 2003. p.121-128.

WILCOX-O'HEARN, Zooko; WARNER, Brian. Tahoe: the least-authority file system. In: ACM INTERNATIONAL WORKSHOP ON STORAGE SECURITY AND SURVIVABILITY, 4., 2008, Alexandria. **Proceedings...** New York: ACM, 2008. p. 21-26.

YU, Weikuan et al. Benefits of high speed interconnects to cluster file systems: a case study with lustre. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED

PROCESSING, 20., 2006. Columbus. **Proceedings...** Washington, DC: IEEE Computer Society, 2006. p.273-280.

APÊNDICE A – Instalação do FlexA desenvolvido

Para a instalação do FlexA desenvolvido é necessária a instalação do interpretador *Python 2.7* (padrão nas distribuições Linux), *netifaces* para administração da interface de rede, *pycrypto* que é responsável por fornecer o conjunto de algoritmos de criptografia, além do Sistema Gerenciador de Banco de Dados SQLite3. Estes componentes citados podem ser instalados através dos comandos abaixo:

```
python2.7
python-dev
python-setuptools
python-utils
python-psutil
sysstat
sqlite3
easy_install pycrypto netifaces
```

Na execução do sistema é necessário o mínimo de 3 servidores primários (devido o envio de 1/3 de porção para cada servidor). No caso dos servidores secundários, são necessários no mínimo 2 servidores, visto que na replicação dois servidores são escolhidos para envio das porções, concretizando a fase de replicação.

Em seguida deve-se fazer o reconhecimento da placa de rede e iniciar o módulo coletor nos servidores secundários:

```
python com.py -r
python coletor_replica.py
```

Após a adequação dos servidores secundários, é necessário iniciar os servidores primários. Inicialmente é feito o reconhecimento da interface de rede, seguido da localização dos servidores secundários, inicialização do `Coletor` - servidor primário, finalizando com a busca dos servidores primários. Estes quatro procedimentos podem ser vistos abaixo:

```
python com.py -r
python com.py -s
python coletor_servidor.py
python com.py -b
```

Após a adequação dos servidores primários, é necessário que os servidores secundários façam a busca pelos servidores primários que foram ativados.

```
python com.py -b
```

Por fim, basta iniciar os clientes para que possam configurar a rede, buscar os servidores primários ativos e iniciar o `Coletor` - cliente para que possa fazer a interação com os demais módulos, descritos abaixo:

```
python com.py -r  
python com.py -b  
python coletor_cliente.py
```

Concluído os procedimentos acima, o FlexA desenvolvido está pronto para as requisições dos usuários, as quais são realizadas através do módulo `flexa.py` [opção] <arquivo> seguido da opção desejada (*put, get, list, delete* ou *new permission*).

APÊNDICE B – Tabelas de tempo

Este Apêndice apresenta tabelas contendo resultados das operações de escrita e leitura de arquivos descritos nas seções que abordam a sobrecarga de servidores primários, a avaliação de sobrecarga dos servidores primários e servidores secundários e comparação das avaliações dos sistemas de arquivos distribuídos FlexA desenvolvido, FlexA original, NFS e Tahoe-LAFS.

A seguir são apresentadas informações relativas ao capítulo 5.5, que abordou a questão da avaliação da sobrecarga de servidores primários e servidores secundários.

As Tabelas 10 e 11 apresentam a porcentagem no consumo de memória nas operações de escrita e leitura de arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB no FlexA desenvolvido de 1 a 32 clientes.

Tabela 10 – Consumo de memória na sobrecarga primário (escrita)

Consumo de Memória na sobrecarga do servidor primário (escrita) (%)						
	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1 Cliente	4	8	9	9	7	7
2 Clientes	4	8	9	9	7	7
4 Clientes	4	9	9	9	8	8
8 Clientes	5	9	9	10	10	12
16 Clientes	6	10	10	12	13	17
32 Clientes	8	11	11	13	26	85

Tabela 11 – Consumo de memória na sobrecarga primário (leitura)

Consumo de Memória na sobrecarga do servidor primário (leitura) (%)						
	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1 Cliente	3	3	3	4	4	4
2 Clientes	3	3	3	4	4	4
4 Clientes	3	3	3	4	4	4
8 Clientes	3	3	4	4	4	4
16 Clientes	3	3	4	4	4	4
32 Clientes	3	3	4	4	4	4

As Tabelas 12 e 13 apresentam a porcentagem da atividade em disco nas operações de escrita e leitura de arquivos no FlexA desenvolvido.

Tabela 12 – Atividade em disco na sobrecarga do servidor primário (escrita)

Atividade em disco na sobrecarga do servidor primário (escrita) (%)						
	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1 Cliente	4	10	11	16	14	15
2 Clientes	5	10	11	16	14	15
4 Clientes	8	9	11	16	14	15
8 Clientes	8	9	15	16	14	15
16 Clientes	8	10	16	17	14	15
32 Clientes	11	12	16	17	16	16

Tabela 13 – Atividade em disco na sobrecarga do servidor primário (leitura)

Atividade em Disco na sobrecarga do servidor primário (leitura) (%)						
	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1 Cliente	0,10	0,54	0,53	0,50	0,49	0,47
2 Clientes	0,11	0,54	0,53	0,50	0,49	0,47
4 Clientes	0,14	0,54	0,52	0,50	0,48	0,47
8 Clientes	0,15	0,53	0,51	0,50	0,48	0,47
16 Clientes	0,15	0,53	0,51	0,49	0,47	0,47
32 Clientes	0,19	0,53	0,51	0,49	0,47	0,46

A seguir são apresentados resultados das avaliações realizadas com os sistemas FlexA original, Tahoe-LAFS, NFS e FlexA desenvolvido, nas operações de escrita e leitura, incluindo os intervalos com 95% de confiança. Tabelas estas, relacionadas ao capítulo 6.

Na Tabela 14 é apresentado o resultado da avaliação na operação de escrita no FlexA original, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB com a utilização que vai de 1 a 16 clientes.

Tabela 14 – Tempos de escrita no FlexA original

Tempos de escrita no FlexA original (segundos)						
Clientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,39	0,63	0,87	1,65	2,96	5,54
IC	0,36 - 0,42	0,62 - 0,64	0,85 - 0,89	1,60 - 1,70	2,93 - 2,99	5,49 - 5,59
2	0,39	0,62	0,88	1,77	3,05	6,55
IC	0,37 - 0,41	0,61 - 0,63	0,86 - 0,90	1,67 - 1,87	2,93 - 3,17	6,15 - 6,95
4	0,38	0,66	0,93	1,80	3,57	7,39
IC	0,36 - 0,40	0,64 - 0,68	0,89 - 0,97	1,72 - 1,88	3,37 - 3,77	6,88 - 7,9
8	0,41	0,68	1,00	1,91	4,02	8,64
IC	0,39 - 0,43	0,66 - 0,70	0,94 - 1,06	1,84 - 1,98	3,83 - 4,21	8,26 - 9,02
16	0,48	0,80	1,21	2,50	6,25	17,33
IC	0,46 - 0,50	0,77 - 0,83	1,16 - 1,26	2,41 - 2,59	5,98 - 6,52	16,80 - 17,86

Na Tabela 15 é apresentado o resultado da avaliação na operação de leitura no FlexA original, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 15 – Tempos de leitura no FlexA original

Tempos de leitura no FlexA original (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	1,11	1,14	1,21	1,49	1,95	4,09
IC	1,04 - 1,18	1,13 - 1,15	1,20 - 1,22	1,45 - 1,53	1,81 - 2,09	3,97 - 4,21
2	1,15	1,13	1,21	1,50	2,03	4,68
IC	1,02 - 1,28	1,13 - 1,13	1,20 - 1,22	1,48 - 1,52	1,91 - 2,15	4,31 - 5,05
4	1,18	1,13	2,23	1,49	2,13	4,63
IC	1,03 - 1,33	1,13 - 1,13	1,83 - 2,63	1,46 - 1,52	2,02 - 2,24	4,42 - 4,84
8	1,12	1,13	1,21	1,44	2,29	4,91
IC	1,04 - 1,20	1,13 - 1,13	1,21 - 1,21	1,43 - 1,45	2,19 - 2,39	4,71 - 5,11
16	1,21	1,13	1,21	1,45	2,57	5,56
IC	1,10 - 1,32	1,13 - 1,13	1,21 - 1,21	1,44 - 1,46	2,28 - 2,46	5,35 - 5,77

Na Tabela 16 é apresentado o resultado da avaliação na operação de escrita no Tahoe-LAFS, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 16 – Tempos de escrita no Tahoe-LAFS

Tempos de escrita no Tahoe-LAFS (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,58	1,39	2,57	7,29	21,88	88,76
IC	0,51 - 0,65	1,38 - 1,40	2,56 - 2,58	7,26 - 7,32	21,84 - 21,92	86,74 - 90,78
2	0,46	0,92	1,57	4,72	12,99	45,42
IC	0,45 - 0,47	0,90 - 0,94	1,54 - 1,60	4,65 - 4,79	12,84 - 13,14	44,91 - 45,93
4	0,45	0,88	1,52	4,54	12,61	44,77
IC	0,44 - 0,46	0,87 - 0,89	1,50 - 1,54	4,43 - 4,65	12,51 - 12,71	44,44 - 45,10
8	0,47	0,90	1,55	4,76	13,73	52,78
IC	0,44 - 0,50	0,89 - 0,91	1,54 - 1,56	4,68 - 4,84	13,50 - 13,96	51,89 - 53,67
16	0,53	1,00	1,67	5,11	17,23	74,68
IC	0,50 - 0,56	0,97 - 1,03	1,63 - 1,71	5,02 - 5,20	16,78 - 17,68	72,59 - 76,77

Na Tabela 17 é apresentado o resultado da avaliação na operação de leitura no Tahoe-LAFS, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 17 – Tempos de leitura no Tahoe-LAFS

Tempos de leitura no Tahoe-LAFS (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,67	0,76	1,29	3,14	8,91	25,86
IC	0,20 - 1,14	0,75 - 0,77	1,28 - 1,30	3,13 - 3,15	8,88 - 8,94	25,83 - 25,89
2	0,39	0,71	1,22	3,73	11,36	47,04
IC	0,38 - 0,40	0,70 - 0,72	1,15 - 1,29	3,36 - 4,10	9,45 - 13,27	44,53 - 49,55
4	0,40	0,72	1,32	4,85	18,35	73,13
IC	0,39 - 0,41	0,70 - 0,74	1,21 - 1,43	4,31 - 5,39	16,74 - 19,96	69,72 - 76,54
8	0,40	0,76	1,77	9,67	42,88	86,16
IC	0,39 - 0,41	0,73 - 0,79	1,57 - 1,97	8,68 - 10,66	38,61 - 47,15	82,47 - 89,85
16	0,91	5,81	25,48	55,27	43,05	107,57
IC	0,79 - 1,03	4,77 - 6,85	21,74 - 29,22	49,12 - 61,42	40,25 - 45,85	104,21 - 110,93

Na Tabela 18 é apresentado o resultado da avaliação na operação de escrita no NFS, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 18 – Tempos de escrita no NFS

Tempos de escrita no NFS (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,12	0,24	0,45	1,10	2,17	3,98
IC	0,11 - 0,13	0,23 - 0,25	0,44 - 0,46	1,09 - 1,11	2,15 - 2,19	3,94 - 4,02
2	0,12	0,23	0,48	1,03	2,12	4,30
IC	0,12 - 0,12	0,22 - 0,24	0,44 - 0,52	0,99 - 1,07	2,02 - 2,22	4,13 - 4,47
4	0,12	0,24	0,47	1,12	2,29	8,31
IC	0,11 - 0,13	0,24 - 0,24	0,44 - 0,50	1,04 - 1,20	2,18 - 2,40	7,54 - 9,08
8	0,12	0,25	0,48	1,63	2,86	18,53
IC	0,12 - 0,12	0,24 - 0,26	0,45 - 0,51	1,52 - 1,74	2,65 - 3,07	17,54 - 19,52
16	0,13	0,27	0,52	1,69	7,89	30,39
IC	0,12 - 0,14	0,26 - 0,28	0,50 - 0,54	1,61 - 1,77	7,49 - 8,29	29,39 - 31,39

Na Tabela 19 é apresentado o resultado da avaliação na operação de leitura no NFS, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 19 – Tempos de leitura no NFS

Tempos de leitura no NFS (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,09	0,18	0,32	0,69	1,32	2,25
IC	0,08 - 0,10	0,17 - 0,19	0,30 - 0,34	0,68 - 0,70	1,29 - 1,35	2,22 - 2,28
2	0,09	0,18	0,30	0,67	1,30	2,25
IC	0,08 - 0,10	0,17 - 0,19	0,29 - 0,31	0,65 - 0,69	1,27 - 1,33	2,22 - 2,28
4	0,10	0,18	0,29	0,62	1,21	2,50
IC	0,09 - 0,11	0,17 - 0,19	0,28 - 0,30	0,61 - 0,63	1,18 - 1,24	2,40 - 2,60
8	0,11	0,18	0,28	0,64	1,20	3,36
IC	0,09 - 0,13	0,17 - 0,19	0,27 - 0,29	0,63 - 0,65	1,18 - 1,22	2,90 - 3,82
16	0,14	0,18	0,28	0,65	1,31	9,72
IC	0,11 - 0,17	0,18 - 0,18	0,27 - 0,29	0,64 - 0,66	1,28 - 1,34	8,03 - 11,41

Na Tabela 20 é apresentado o resultado da avaliação na operação de escrita no FlexA desenvolvido, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 20 – Tempos de escrita no FlexA desenvolvido

Tempos de escrita no FlexA desenvolvido (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	0,08	0,22	0,42	0,97	1,93	3,94
IC	0,08 - 0,08	0,21 - 0,23	0,40 - 0,44	0,92 - 1,2	1,89 - 1,97	3,83 - 4,05
2	0,10	0,23	0,43	0,98	1,97	4,06
IC	0,09 - 0,11	0,21 - 0,25	0,41 - 0,45	0,95 - 1,01	1,92 - 2,02	3,89 - 4,23
4	0,12	0,24	0,46	1,01	2,04	4,31
IC	0,10 - 0,14	0,22 - 0,26	0,43 - 0,49	0,98 - 1,04	1,96 - 2,12	4,08 - 4,54
8	0,13	0,26	0,50	1,11	2,48	5,12
IC	0,12 - 0,14	0,24 - 0,28	0,48 - 0,52	1,07 - 1,15	2,36 - 2,60	4,87 - 5,37
16	0,17	0,30	0,55	1,29	2,84	5,99
IC	0,15 - 0,19	0,28 - 0,32	0,52 - 0,58	1,23 - 1,35	2,74 - 2,94	5,80 - 6,18

Na Tabela 21 é apresentado o resultado da avaliação na operação de leitura no FlexA desenvolvido, utilizando arquivos de 1 MB, 5 MB, 10 MB, 25 MB, 50 MB e 100 MB.

Tabela 21 – Tempos de leitura no FlexA desenvolvido

Tempos de leitura no FlexA desenvolvido (segundos)						
Cientes	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
1	1,09	1,13	1,20	1,37	2,44	3,62
IC	1,09 - 1,09	1,13 - 1,13	1,20 - 1,20	1,36 - 1,38	2,37 - 2,51	3,55 - 3,69
2	1,09	1,13	1,21	1,37	2,49	4,00
IC	1,09 - 1,09	1,13 - 1,13	1,21 - 1,21	1,37 - 1,37	2,44 - 2,54	3,65 - 4,35
4	1,17	1,13	1,20	1,37	2,54	4,58
IC	1,15 - 1,19	1,13 - 1,13	1,20 - 1,20	1,37 - 1,37	2,48 - 2,60	4,23 - 4,93
8	1,09	1,13	1,20	1,37	2,65	4,96
IC	1,08 - 1,10	1,13 - 1,13	1,20 - 1,20	1,37 - 1,37	2,55 - 2,75	4,68 - 5,24
16	1,12	1,13	1,21	1,38	2,37	4,57
IC	1,10 - 1,14	1,13 - 1,13	1,21 - 1,21	1,38 - 1,38	2,53 - 2,61	4,38 - 4,76

Por fim, são apresentadas tabelas comparativas com tempos nas operações de escrita e leitura utilizando os sistemas abordados anteriormente.

Na Tabela 22 é apresentado comparativo nas operações de escrita e leitura a partir de 1 cliente.

Tabela 22 – Comparação de tempos na escrita e leitura por 1 cliente

Comparação nos tempos de escrita – 1 cliente (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	0,39	0,63	0,87	1,65	2,96	5,54
FlexA desenvolvido	0,08	0,22	0,42	0,97	1,93	3,94
NFS	0,12	0,24	0,45	1,10	2,17	3,98
Tahoe-LAFS	0,58	1,39	2,57	7,29	21,88	88,76
Comparação nos tempos de leitura – 1 cliente (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	1,11	1,14	1,21	1,49	1,95	4,09
FlexA desenvolvido	1,09	1,13	1,20	1,37	2,44	3,62
NFS	0,09	0,18	0,32	0,69	1,32	2,25
Tahoe-LAFS	0,67	0,76	1,29	3,14	8,91	25,86

Na Tabela 23 é apresentado comparativo nas operações de escrita e leitura a partir de 2 cliente.

Tabela 23 – Comparação de tempos na escrita e leitura por 2 clientes

Comparação nos tempos de escrita – 2 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	0,39	0,62	0,88	1,77	3,05	6,55
FlexA desenvolvido	0,10	0,23	0,43	0,98	1,97	4,06
NFS	0,12	0,23	0,48	1,03	2,12	4,30
Tahoe-LAFS	0,46	0,92	1,57	4,72	12,99	45,42
Comparação nos tempos de leitura – 2 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	1,15	1,13	1,21	1,50	2,03	4,68
FlexA desenvolvido	1,09	1,13	1,21	1,37	2,49	4,00
NFS	0,09	0,18	0,30	0,67	1,30	2,25
Tahoe-LAFS	0,39	0,71	1,22	3,73	11,36	47,04

Na Tabela 24 é apresentado comparativo nas operações de escrita e leitura a partir de 4 clientes.

Tabela 24 – Comparação de tempos na escrita e leitura por 4 clientes

Comparação nos tempos de escrita – 4 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	0,38	0,66	0,93	1,80	3,57	7,39
FlexA desenvolvido	0,12	0,24	0,46	1,01	2,04	4,31
NFS	0,12	0,24	0,47	1,12	2,29	8,31
Tahoe-LAFS	0,45	0,88	1,52	4,54	12,61	44,77
Comparação nos tempos de leitura – 4 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	1,18	1,13	1,25 – 2,23	1,49	2,13	4,63
FlexA desenvolvido	1,17	1,13	1,20	1,37	2,54	4,58
NFS	0,10	0,18	0,29	0,62	1,21	2,50
Tahoe-LAFS	0,40	0,72	1,32	4,85	18,35	73,13

Na Tabela 25 é apresentado comparativo nas operações de escrita e leitura a partir de 8 clientes.

Tabela 25 – Comparação de tempos na escrita e leitura por 8 clientes

Comparação nos tempos de escrita – 8 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	0,41	0,68	1,00	1,91	4,02	8,64
FlexA desenvolvido	0,13	0,26	0,50	1,11	2,48	5,12
NFS	0,12	0,25	0,48	1,63	2,86	18,53
Tahoe-LAFS	0,47	0,90	1,55	4,76	13,73	52,78
Comparação nos tempos de leitura – 8 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	1,12	1,13	1,21	1,44	2,29	4,91
FlexA desenvolvido	1,09	1,13	1,20	1,37	2,65	4,96
NFS	0,11	0,18	0,28	0,64	1,20	3,36
Tahoe-LAFS	0,40	0,76	1,77	9,67	42,88	86,16

A Tabela 26 apresenta comparativo nas operações de escrita e leitura a partir de 16 clientes.

Tabela 26 – Comparação de tempos na escrita e leitura por 16 clientes

Comparação nos tempos de escrita – 16 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	0,48	0,80	1,21	2,50	6,25	17,33
FlexA desenvolvido	0,17	0,30	0,55	1,29	2,84	5,99
NFS	0,13	0,27	0,52	1,69	7,89	30,39
Tahoe-LAFS	0,53	1,00	1,67	5,11	17,23	74,68
Comparação nos tempos de leitura – 1 clientes (segundos)						
Tipo SAD	1 MB	5 MB	10 MB	25 MB	50 MB	100 MB
FlexA original	1,21	1,13	1,21	1,45	2,57	5,56
FlexA desenvolvido	1,12	1,13	1,21	1,38	2,37	4,57
NFS	0,14	0,18	0,28	0,65	1,31	9,72
Tahoe-LAFS	0,91	5,81	25,48	55,27	43,05	107,57