Quinta Conferencia de Directores de Tecnología de Información y Comunicación en Instituciones de Educación Superior: Gestión de las TICs para la investigación y colaboración

Uma proposta de arquitetura para melhoria de desempenho no ambiente virtual de aprendizagem Moodle utilizando proxy reverso

Sérgio Sant'Anna de Sá [a], Saymon Castro de Souza [a],

[a] Instituto Federal do Espírito Santo, Rodovia ES-010, Km 6.5,
29173-087 Serra, Brasil
sergio.sa@ifes.edu.br, saymon@ifes.edu.br

Resumo. Este trabalho propõe uma arquitetura que utiliza um proxy reverso como forma de melhorar o desempenho do ambiente virtual de aprendizagem Moodle. Foram executados testes simulando acessos de 50, 100, 150 e 200 usuários simultâneos comparando um cenário cliente servidor tradicional e a arquitetura com proxy reverso. Os resultados mostraram que houve uma melhora significativa na latência dos objetos que foram configurados para serem armazenados no Proxy reverso. Finalmente, são realizadas algumas considerações em relação à adoção da arquitetura proposta.

Palavras-Chave: Moodle, EAD, ambiente virtual de aprendizagem, proxy reverso.

1 Introdução

O Centro de Referência em Formação e em Educação a Distância – Cefor [1] do Instituto Federal do Espírito Santo - Ifes [2] é responsável pela institucionalização da Educação a Distância (EAD) e pela formação inicial e continuada de professores e técnicos administrativos da educação, além da infraestrutura tecnológica para a EAD. Os cursos do Ifes na modalidade a distância foram inicialmente ofertados pelo Cefor e, em 2011, vinculados aos campi responsáveis pela elaboração do projeto pedagógico dos cursos. Atualmente o Cefor apoia dois cursos técnicos, quatro graduações, seis pós-graduações, além dos cursos de formação de tutores, professores e designers instrucionais. Dos 78 municípios do estado Espírito Santo, 35 são atendidos pelos cursos EAD oferecidos pela instituição. Além dos cursos EAD, os 20 campi do Ifes também utilizam a infraestrutura tecnológica do Cefor como forma de apoio aos cursos presenciais. De aproximadamente 16.200 alunos do Ifes no ano de 2014, cerca de 3.200 (20%) eram da EAD.

A educação a distância tem um papel social importante, pois devido às características de rompimento temporal e espacial, o acesso é facilitado para as

pessoas que tem dificuldade de frequentar a modalidade presencial. Segundo [3], a Educação a Distância é o "processo de ensino-aprendizagem, mediado por tecnologias, onde professores e alunos estão separados espacial e/ou temporalmente".

As plataformas de educação a distância utilizadas por meio da internet são chamadas de AVA - Ambiente Virtual de Aprendizagem, uma adaptação do inglês *Learning Management System* (LMS) [5]. Existem diversas implementações de ambiente virtuais de aprendizagem, como o TelEduc [6], Eureka [7] e Moodle [8], este último é o AVA utilizado pelo Cefor/Ifes.

De acordo com [4] a educação a distância no Brasil apresenta um crescimento exponencial e que há um aumento de interesse da sociedade em adquirir conhecimento. Dessa forma, o crescimento da EAD e a popularização da utilização dos ambientes virtuais de aprendizagem, tais como o Moodle, tem um impacto direto nas tecnologias da informação e comunicação que mediam o processo de ensino-aprendizagem. Tal impacto eleva a importância de alguns requisitos de sistemas web, em especial atenção para o desempenho e a disponibilidade, no contexto deste trabalho.

Existem diversos trabalhos que propõem abordagens para melhoria do desempenho e disponibilidade em sistemas web, como em [9] onde é apresentado um mecanismo de controle de admissão e balanceamento de carga em clusters de servidores web, onde conta com parâmetros para utilizar de forma eficaz os recursos de processamento. Em [10] é apresentado um algoritmo de cache inteligente, capaz de adaptar seu comportamento com base em estatísticas de acesso. Em [11] é apresentado um algoritmo de substituição de objetos em cache na internet. Trata-se de um problema que já vem sendo estudado na literatura há algum tempo, dentre as diversas propostas para atenuar o problema supracitado, é possível destacar o *web caching*, que se refere a uma técnica para aliviar o gargalo no servidor e reduzir o tráfego de rede, assim, minimizando a latência no acesso dos usuários [12]. O proxy reverso é um tipo de serviço de *web caching*, onde o cache é implementado perto da origem do conteúdo, oposto ao cache implementado do lado do cliente. Esta solução é atraente para os servidores que esperam um elevado número de requisições e querem garantir um alto nível de qualidade de serviço [13].

Portanto, para lidar com o crescimento da EAD no país e manter níveis satisfatórios de desempenho e disponibilidade de um ambiente virtual de aprendizagem, empregar técnicas de *web caching* se mostram promissoras para atender a necessidade tecnológica dos requisitos supracitados.

2 Fundamentação teórica

Este trabalho explora a implementação de um proxy reverso, como forma de melhoria de desempenho de um ambiente virtual de aprendizagem. Para um melhor entendimento das contribuições do trabalho, esta seção realiza uma revisão dos conceitos básicos dessas áreas de pesquisa.

2.1 Proxy

O proxy (também conhecido como proxy de encaminhamento) é o servidor responsável por realizar solicitações HTTP para os clientes de uma rede privada. Se um objeto solicitado não for encontrado no cache, o proxy deve realizar uma solicitação em nome do cliente para o servidor de origem. Após obter o objeto do servidor de origem, uma cópia do mesmo pode ser armazenada no cache antes de encaminhá-lo ao cliente. Nessa abordagem, todos os clientes HTTP devem ser configurados manualmente para utilizar o servidor de proxy, o que pode tornar inviável sua implantação em um ambiente com muitos clientes [14]. A Fig. 1 apresenta essa arquitetura.

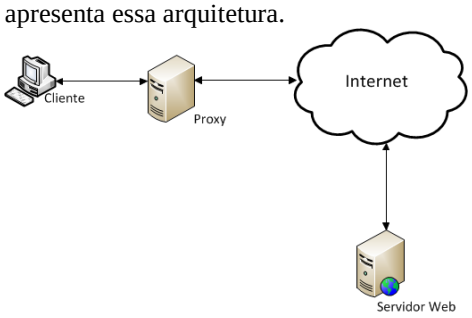

Fig. 1. Arquitetura de um proxy.

Nas subseções seguintes (2.2 e 2.3) são apresentados duas variações de arquiteturas de proxy, a saber transparent caching e proxy reverso; e a subseção 2.4 apresenta a implementação de proxy reverso utilizada na arquitetura proposta neste trabalho.

2.2 Transparent caching

O *transparent caching*, também chamado de proxy transparente [12], elimina o grande inconveniente da configuração manual dos clientes HTTP na abordagem de proxy, já que ele intercepta as requisições HTTP na porta 80 e as encaminha para servidores de web cache ou cluster cache. A vantagem desta abordagem também pode ser considerada sua fraqueza, pois ele viola o argumento de conexão fim-a-fim por não manter constante os estados de conexão. Isso pode gerar problemas com aplicativos que exigem que o estado da conexão fique ativo ao longo de pedidos consecutivos. Outra desvantagem é que interceptar todo o tráfego direcionado para a porta 80 acrescenta uma latência adicional [13] [14].

2.3 Proxy reverso

O proxy reverso, apresentado na Fig. 2, é uma variação do proxy de encaminhamento, onde o cache fica próximo a origem do conteúdo (servidor de origem) ao invés de ficar próximo do cliente. Essa variação de proxy permite que o servidor antecipe uma grande quantidade de requisições para que possa manter uma qualidade superior de

serviço, além de ser totalmente autônoma e transparente para os clientes, já que não precisam de nenhum tipo de configuração, ao contrário dos clientes do proxy de encaminhamento [14].

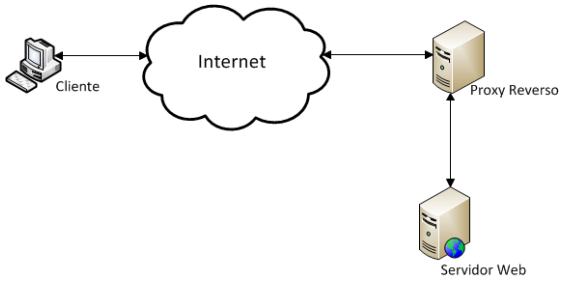

Fig. 2. Arquitetura de um proxy reverso

2.4 Varnish

O Varnish é uma implementação de proxy reverso, que desde o início, foi projetado como tal, diferente de outras implementações como o Squid, que inicialmente foram projetados para trabalhar como proxy de encaminhamento, ou seja, do lado do cliente. Este é um projeto *open-source*, baseado e testado em GNU/Linux e FreeBSD [16].

Esta implementação de proxy reverso possui algumas características relevantes, tais como: flexibilidade e desempenho. A flexibilidade é devido a sua linguagem específica de domínio (DSL – *domain-specific language*) chamada de VCL (*Varnish Configuration Language*), onde é possível definir diversas políticas de cache, como o servidor web no qual o Varnish deve buscar os arquivos, por quanto tempo e quais arquivos que devem ser armazenados no cache, entre outros. Segundo alguns experimentos, dependendo da arquitetura, é possível acelerar as entregas de conteúdo entre 300 e 1000 vezes [16].

O arquivo padrão com as políticas é armazenado em *"/etc/varnish/default.vcl"*. Os arquivos de configuração VCL são divididos em "funções", na qual são executadas em momentos diferentes. O conceito "função" do VCL é diferente se comparado à linguagens de programação, pois as declarações de retorno de cada "função" são chamadas para outra "função" do Varnish. Algumas das funções do Varnish são detalhadas abaixo:

- *backend*: trata-se da indicação do(s) servidor(es) que proveem o serviço onde Varnish deve buscar conteúdo, tais servidores devem suportar o protocolo HTTP. É possível configurar mais de um *backend* e há a possibilidade de realizar balanceamento de carga entre eles.
- *vcl_recv*: é chamado no início de uma solicitação HTTP, após ela ser recebida e analisada por completo. Seu objetivo é decidir se deve ou não atender à solicitação, como atender, e se for o caso, qual *backend* utilizar. Nessa função é possível alterar a solicitação HTTP, normalmente os *cookies* e os cabeçalhos.

- *vcl_fetch*: é acionado depois que um documento foi recuperado com êxito do *backend*. Tarefas normais desta função são alterar o cabeçalho da resposta ou tentar servidores alternativos em caso de falha na solicitação, quando existe um balanceamento de carga configurado.
- *vcl_hash*: é utilizada para criar um hash do objeto a ser armazenado pelo Varnish.
- *O return()* é utilizado quando a execução de uma função está concluída, porém um número limitado deles estão disponíveis em cada função. Os principais retornos são:

    - *pass*: a solicitação é repassada para o backend.
    - *hist_for_pass*: o conteúdo não deve ser copiado para o cache.
    - *lookup*: o conteúdo deve ser buscado no cache.
    - *deliver*: entrega o objeto em cache para o cliente (VARNISH, 2014).

O Varnish é uma implementação de proxy reverso que se mostra promissora para melhorar o desempenho e disponibilidade de acessos simultâneos em um ambiente virtual de aprendizagem.

2.5 Moodle

O Moodle [8] é uma plataforma de aprendizagem *open source*, estruturada em torno de cursos - páginas onde os professores disponibilizam os recursos e atividades aos alunos. Os recursos tem a função de apresentar o conteúdo do curso aos alunos, dentre os quais é possível citar: páginas de textos simples, links para outros sites, arquivos para download entre outros. Já as atividades (ex.: questionários, fóruns, tarefas, wikis, glossários etc) são ferramentas de avaliação ou comunicação com os alunos [8]. Os conteúdos são exibidos nas sessões centrais e os blocos laterais oferecem recursos extras, como as opções de configuração do curso dentre outros, conforme a Fig. 3.

Fig. 3. Um curso no Moodle do Cefor/Ifes na visão de administrador. Entre as linhas vermelhas a sessão central e os blocos laterais entre as linhas amarelas.

Tipicamente, os recursos são componentes estáticos do curso, ou seja, raramente são modificados durante a execução do curso. As atividades podem ser definidas como elementos dinâmicos, ou seja, frequentemente ocorrem atualizações de seus conteúdos. Essa classificação é relevante para uma análise do Moodle em relação à abordagem proposta neste artigo.

3 Análise do ambiente virtual de aprendizagem Moodle

A caracterização do comportamento de acesso dos usuários é um fator importante para adequação da arquitetura proposta com intuito de melhorar o desempenho do acesso ao Moodle. Um comportamento recorrente de acesso ao curso, atividades e recursos foi observado por meio de consultas ao banco de dados do Moodle – Cefor/Ifes: o aluno realiza o acesso à página principal do curso, navega entre os recursos e/ou atividades, eventualmente retornando a página principal do curso, em um processo cíclico.

O recurso, por raramente sofrer alterações e possuir muitos acessos, se torna um elemento importante para a melhoria de desempenho. A Fig. 4 apresenta um exemplo registros de acesso de um recurso, no formato PDF, em um curso no Moodle do Cefor/Ifes. São diversos os horários em que os alunos acessaram este recurso, porém o comportamento de acesso referente ao aspecto temporal dos cursos e até mesmo dos recursos pode variar. O rompimento espaço-temporal, uma das características marcantes na EaD, influencia no aspecto tecnológico. Esse comportamento de acesso aos cursos pode contribuir no processo de análise dos parâmetros temporais do *cache*.

Hora	Endereço IP	Nome completo	Ação	Informação
dom 27 abril 2014, 18:53	1	Jé	resource view	Material da 1ª Semana
dom 27 abril 2014, 18:38	1	Si	resource view	Material da 1ª Semana
dom 27 abril 2014, 18:11	2	Wa	resource view	Material da 1ª Semana
dom 27 abril 2014, 17:50	1	So	resource view	Material da 1ª Semana
dom 27 abril 2014, 17:22	1	An	resource view	Material da 1ª Semana
dom 27 abril 2014, 17:07	2	Br	resource view	Material da 1ª Semana
dom 27 abril 2014, 17:07	2	Lo	resource view	Material da 1ª Semana
dom 27 abril 2014, 16:34	1	Th	resource view	Material da 1ª Semana
dom 27 abril 2014, 16:18	1	Si	resource view	Material da 1ª Semana
dom 27 abril 2014, 15:29	1	Al	resource view	Material da 1ª Semana
dom 27 abril 2014, 15:29	1	Al	resource view	Material da 1ª Semana
dom 27 abril 2014, 15:01	1	Ma	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:35	1	Se	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:34	1	Le	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:30	1	Le	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:30	1	Le	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:30	1	Le	resource view	Material da 1ª Semana
dom 27 abril 2014, 14:20	1	Ma	resource view	Material da 1ª Semana
dom 27 abril 2014, 13:51	1	Ma	resource view	Material da 1ª Semana
dom 27 abril 2014, 13:46	1	Ro	resource view	Material da 1ª Semana
dom 27 abril 2014, 12:36	1	Mi	resource view	Material da 1ª Semana
dom 27 abril 2014, 12:22	1	Th	resource view	Material da 1ª Semana

Fig. 4. Relatório de acesso de um arquivo PDF de um curso no Moodle do Cefor/Ifes.

A página inicial de um curso no Moodle tende a não sofrer mudanças frequentes, pois os cursos da modalidade a distância do Ifes são planejados e revisados previamente, sendo assim, é também um elemento promissor para melhoria de desempenho.

Com vários usuários acessando o ambiente simultaneamente, existe uma tendência de aumento o uso do processador, memória RAM, acesso ao disco e tráfego de rede do servidor, além das demandas de leitura e escrita no banco de dados. Com o aumento dessas demandas, a latência de resposta do servidor Web tende a aumentar. Como forma de melhorar o desempenho do servidor Web, que hospeda um AVA, este trabalho propõe que alguns elementos do AVA sejam armazenados em *cache* sem que hajam prejuízos aos alunos, já que os elementos escolhidos tendem a ser alterados com pouca frequência.

Portanto, conforme a discussão supracitada, foram definidos os elementos a serem armazenados em *cache* com intuito de melhorar o desempenho do sistema:

- Páginas iniciais dos cursos;
- Páginas de recursos;
- Arquivos pdf e doc.

Um aspecto importante se refere à sessão que o Moodle utiliza para os usuários autenticados [8]. Caso o elemento apresente informações personalizadas, como o nome do usuário autenticado no ambiente, é necessária a criação de um *cache* para cada usuário autenticado no Moodle. A próxima seção discute detalhes da arquitetura proposta neste artigo.

4. Arquitetura

A arquitetura proposta neste trabalho é a utilização de um proxy reverso, com a finalidade de melhorar o desempenho do ambiente virtual de aprendizagem, por meio da utilização do cache para elementos do sistema que tem pouca modificação, conforme discutido na seção anterior. A aplicação de proxy reverso Varnish foi implementada em conjunto com o Moodle. A Fig. 5 apresenta, de forma geral, a arquitetura implementada.

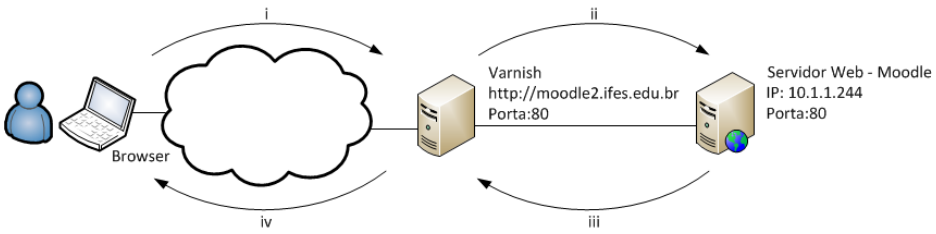

Fig. 5. Arquitetura do Varnish integrada ao Moodle

O Varnish é responsável por receber as requisições, encaminhá-las ao servidor web e responder os usuários. Todas as solicitações que chegam dos usuários são tratadas de acordo com o bloco da função *vcl_recv (i)*, presente no arquivo de configuração das políticas de *cache*. Caso não possua a cópia do arquivo solicitado pelo usuário, o Varnish deve realizar uma solicitação ao servidor web através de uma requisição HTTP (ii). Se o Varnish possuir a cópia do arquivo, ele deve ser tratado e enviado para o usuário, conforme iv. O servidor web recebe a requisição, processa e envia

uma mensagem de resposta HTTP para o Varnish. Caso a página solicitada for encontrada no cache, ela é enviada para o usuário. Após receber as respostas das solicitações enviadas para o servidor web, o Varnish deve realizar o tratamento dessa resposta de acordo com o arquivo de configuração das políticas de *cache*, no bloco da função *vcl_fetch*. Após o tratamento da reposta, ela é enviada para o usuário.

O arquivo com as políticas de cache definidas para o escopo deste trabalho está disponível em https://github.com/sergiosantanna/varnish.

5 Testes de desempenho

A infraestrutura física utilizada nos testes é composta por um cluster de alta disponibilidade, com três hosts e um storage, executando VMWare 5.5. Foram criadas duas máquinas virtuais; a primeira para prover o Moodle 2.6.4+ e outra para o Varnish 3.05, ambas utilizando o sistema operacional Linux. Seguem as respectivas configurações de hardware:

- 4 CPUs Intel "Westmere" Genaration, 32GB RAM, 50GB Disco virtual;
- 4 CPUs Intel "Westmere" Genaration, 16GB RAM, 50GB Disco virtual.

O Moodle possui mecanismos de auditoria, no qual todas as ações do usuário no ambiente, como informações de login, as páginas acessadas, visualização, adição e modificação de recursos e atividades, participação em fóruns, entre outros são registrados em seu banco de dados.

Foi realizada uma consulta no banco de dados do Moodle do Cefor/Ifes com intuito de caracterizar o comportamento de acesso dos usuários no ambiente. Nessa consulta, foram escolhidas duas salas virtuais aleatórias, de cada um dos níveis de ensino: técnico, graduação e pós-graduação. O resultado da consulta é apresentado na Fig. 6.

Página	Ação	Acessos	Percentual de acesso
course	Acessar	47745	26,41%
forum	Acessar discussão	30073	16,63%
forum	Acessar fórum	28590	15,81%
resource	Acessar	17849	9,87%
assignment	Acessar	10718	5,93%
wiki	Acessar	5631	3,11%
quiz	Continuar tentativa	5513	3,05%
user	Acessar todos	5300	2,93%
quiz	Acessar	4712	2,61%
user	Acessar	3722	2,06%
forum	Adicionar postagem	2071	1,15%
Outros	Outros	< 2071	< 1%
Total de acessos		180805	100,00%

Fig. 6. Resultado da consulta do comportamento dos usuários do Moodle Cefor/Ifes

A finalidade da composição do quadro acima é identificar as ações realizadas pelos usuários e apresentar o percentual em relação ao total de acessos. Essas informações

subsidiaram a maneira em que foram configuradas as simulações de acesso ao ambiente, com intuito de torna-los mais fidedignas aos acessos existentes do Moodle Cefor/Ifes.

Dois casos de teste foram definidos: SemProxy: os usuários tem acesso direto ao Moodle (Fig. 7); ComProxy: os usuários tem o servidor de proxy reverso Varnish como intermediário, no qual encaminha as requisições para o servidor do Moodle de forma totalmente transparente (Fig. 8). Ambos os casos de teste foram executados com quatro grupos de usuários, a saber: 50, 100, 150 e 200 simultâneos. A quantidade de usuários foram definidos com base nos valores registrados pelo Google Analitycs [19], em torno de 50 a 200 usuários simultâneos.

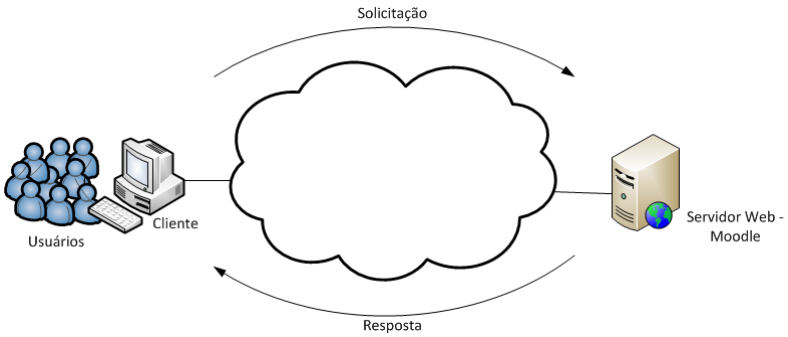

Fig. 7. Arquitetura cliente-servidor.

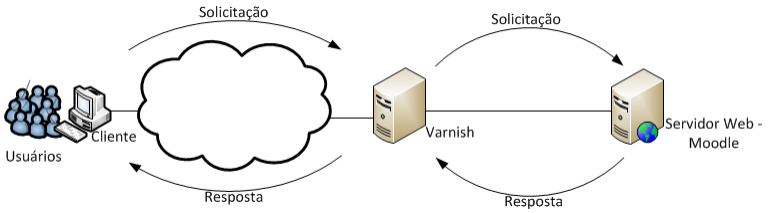

Fig. 8. Arquitetura cliente-servidor com proxy reverso.

O plano de testes foi definido baseado no comportamento dos usuários conforme o quadro supracitado (Fig. 6). A ferramenta JMeter [17] foi utilizada para executar as simulações de acesso ao Moodle, ela fornece uma estrutura na qual é possível definir a execução de uma sequência de operações ou ferramentas. A Fig. 9 apresenta o plano de testes criado no JMeter.

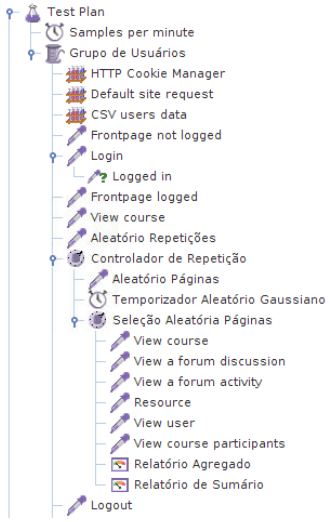

Fig. 9. Requisições no plano de teste do JMeter

Segue abaixo a descrição detalhada do plano de testes apresentado na Fig. 9.

- Test Plan: são configurados o endereço e caminho do servidor, além de páginas a serem acessadas durante a execução dos testes;
- Grupo de usuários: é definido quantidade de usuários virtuais utilizados no teste. Dentro dessa configuração de grupos de usuários estão todas as samplers (requisições) a serem realizados no teste;
- Cookie Manager: A primeira opção configurada após o grupo de usuários, utilizado para armazenar e enviar cookies da mesma forma que um navegador Web [18]. É utilizado pois o Moodle utiliza cookies para armazenar informações de sessão e manter os usuário autênticados no ambiente [8]. Dessa forma, cada usuário virtual do JMeter terá sua própria sessão no Moodle durante a execução do plano de testes;
- Default site request: deve ser configurado o endereço do servidor e do path do site testado;
- CSV users data: O caminho de um arquivo CSV deve ser informado nessa opção, sendo que tal arquivo deve conter os nomes de usuários e senha de usuários cadastrados no ambiente a ser testado;
- Frontpage not logged: iniciando a sequência de acessos ao Moodle, a página a ser acessar é a página inicial antes de o login ser realizado;
- Login: em seguida a página de login é acessada e cada usuário se autentica no ambiente;
- Frontpage logged: página inicial do Moodle é novamente acessada;
- View course: acessa a página inicial do curso utilizado no plano de testes;
- Aleatório Repetição: testador BeanShell, executados antes da execução de cada requisição devidamente configurada pra utilizar o valor de

retorno desse testador [18].Um número entre 0 e 50 é gerado para que cada usuário repita a quantidade de acessos aos recursos e atividades do Moodle igual ao número gerado;

- Controlador de repetição: cada usuário vai repetir o acesso às páginas de acordo com o número gerado em Aleatório Repetição;
    - Aleatório Página: outro testador BeanShell, onde um número entre 0 e 5 é gerado para que o acesso aos recursos e atividades do Moodle seja realizado de forma aleatória, com uma configuração de pesos para simular os acesso conforme os resultados do comportamento dos usuários no Moodle do Cefor/Ifes, conforme apresentado na Fig. 6;
    - Temporizado Aleatório Gaussiano: é uma pausa com o período de tempo aleatório antes de a requisição ser realizada [18], utilizada para simular o thinking time do usuário;
    - Seleção Aleatória Páginas: o usuário realiza de forma aleatória, uma das páginas configuradas nesse controlador. Foram configuradas 6 opções de páginas a serem acessadas, cada uma possui um número de sequência, iniciando no 0. O número aleatório entre 0 e 5 gerado em Aleatório Páginas é utilizado nesse controlador para executar cada requisição de forma aleatória a cada acesso do usuário. A seguir a descrição das páginas que podem ser acessadas;
        - View course: página inicial do curso utilizado no plano de testes;
        - View a forum discussion: entrar no fórum de discussão;
        - View a forum activity: visualizar as atividades recentes do fórum;
        - Resource: o recurso desse curso, a saber, um arquivo PDF de 1.1MB;
        - View user: exibir o perfil de um usuário e, por fim;
        - View a course participants: exibir a lista de participantes do curso.
- Logout: a última requisição a ser realizada, após todas as repetições de cada usuário encerrar, é realizar o logout no Moodle.

6 Resultados

Durante o processo de execução do plano de testes, o JMeter produz um arquivo de log que contém informações dos parâmetros avaliados, como o tempo de resposta médio, mínino e máximo, taxa de erro e desvio padrão das requisições realizadas. Tais arquivos foram utilizados como fonte de dados para gerar os gráficos da latência média das páginas acessadas durante a execução do plano de teste.

Conforme apresentado na seção anterior, foram realizados diversos testes utilizando os dois casos de testes, ambos utilizando 50, 100, 150 e 200 usuários: SemProxy - os acessos são realizados diretamente no servidor web e; ComProxy - os acessos dos usuários são realizados com intermédio do proxy reverso.

Durante o caso de teste ComProxy, quando cada usuário virtual do JMeter realizou o primeiro acesso à página do curso, ela não foi encontrada no cache do Varnish, que então, encaminhou a solicitação da página para o servidor web no qual o Moodle foi configurado. Após receber a página do servidor web, o Varnish armazenou a página no cache em seguida a entregou para o usuário que realizou a solicitação. No acesso seguinte a essa página, o Varnish realizou a entrega sem a necessidade de consultar o servidor web, pois a página estava armazenada no cache.

Já o arquivo PDF, após ser solicitado pela primeira vez durante o caso de teste ComProxy, o Varnish não o encontrou no cache e então realizou uma consulta ao *backend*, e, após receber a solicitação, fez uma cópia do arquivo no cache e em seguida enviou ao usuário que o solicitou. Após todos os acessos seguintes não houve necessidade do Varnish consultar o backend, pois o arquivo estava armazenado no cache.

As Fig. 10, Fig. 11, Fig. 12 e Fig. 13 apresentam os resultados dos testes executados em gráficos de colunas, comparando a latência média da requisição de cada elemento do Moodle, em ambos os casos de teste executados. É possível observar que a latência média das páginas configuradas para serem armazenadas em cache foram menores em todos os cenários testados. Já nos elementos do Moodle onde o armazenamento no proxy não foi configurado, é possível afirmar que a diferença entre a latência média no caso de teste ComProxy e do caso de teste SemProxy é similar (Fig. 16, Fig. 17, Fig. 18, Fig. 19), logo, a hipótese que a presença do Proxy reverso poderia influenciar negativamente em um contexto de *cache miss*, devido a necessidade de atuação como um intermediário entre o usuário e o servidor web, pode ser considerada falsa.

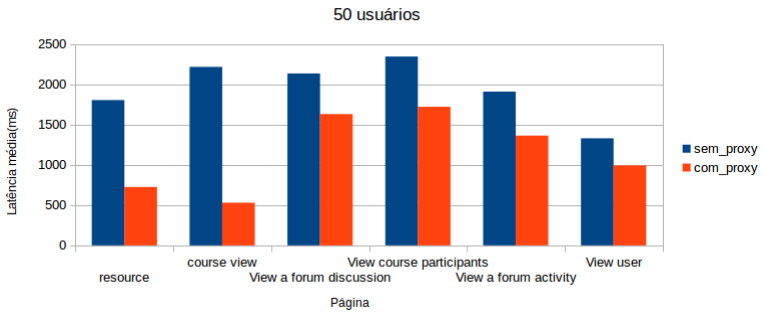

Fig. 10. Latência média com 50 usuários.

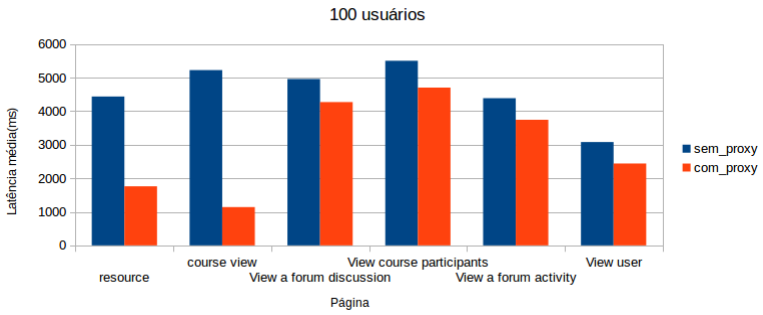

Fig. 11. Latência média com 100 usuários.

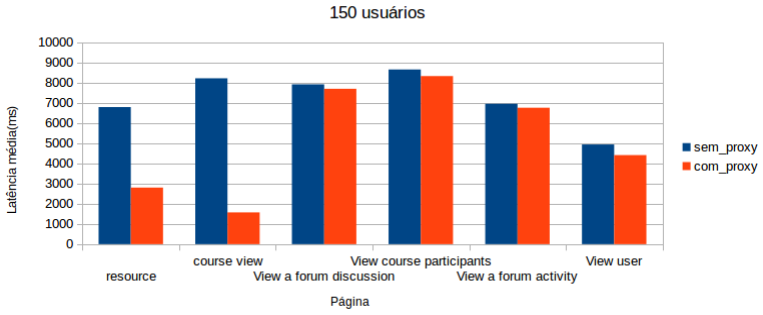

Fig. 12. Latência média com 150 usuários.

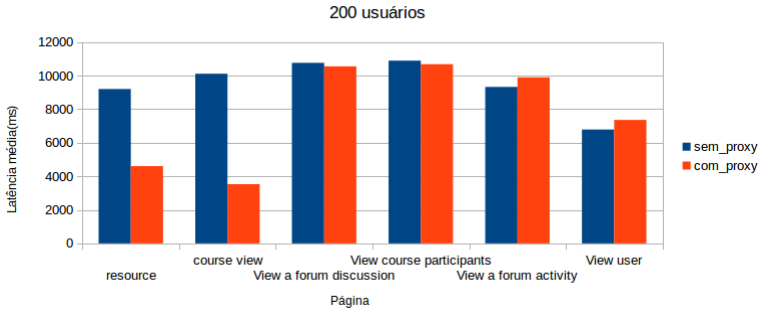

Fig. 13. Latência média com 200 usuários.

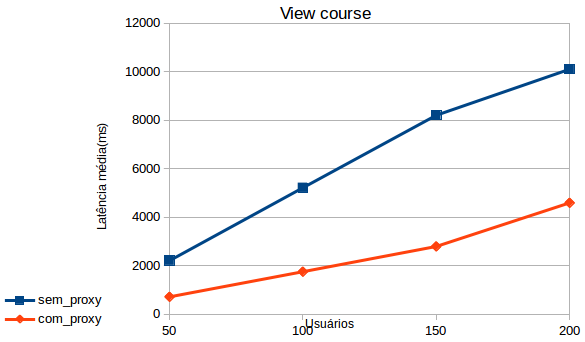

Fig. 14. Latência média da requisição View course de todos usuários.

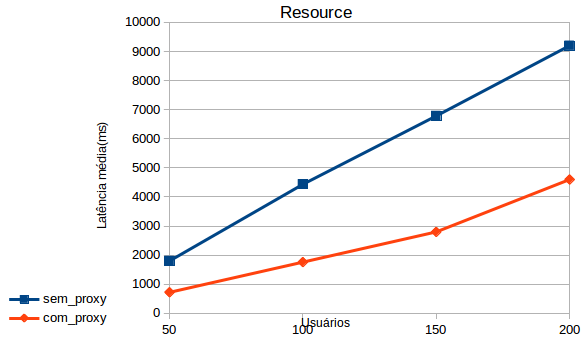

Fig. 15. Latência média da requisição Resource de todos os usuários.

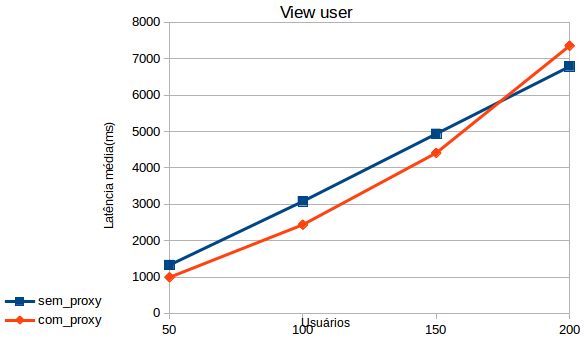

Fig. 16. Latência média da requisição View user de todos os usuários.

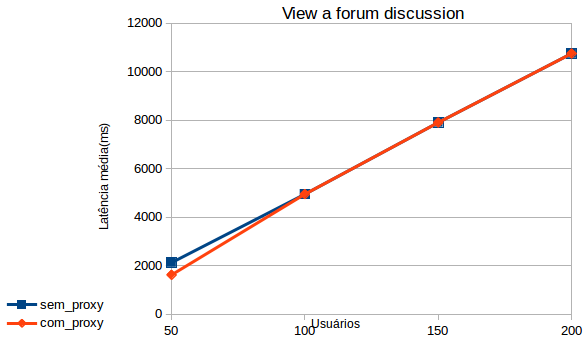

Fig. 17. Latência média da requisição View a forum discussion de todos os usuários.

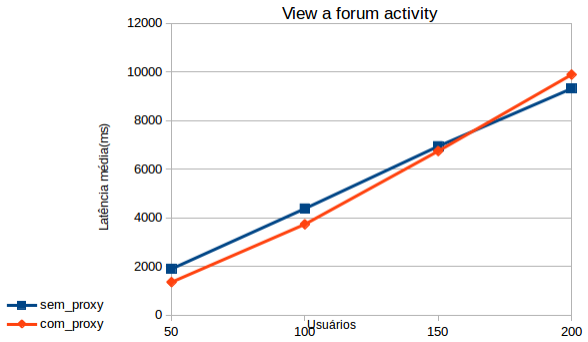

Fig. 18. Latência média da requisição View a fórum activity de todos os usuários.

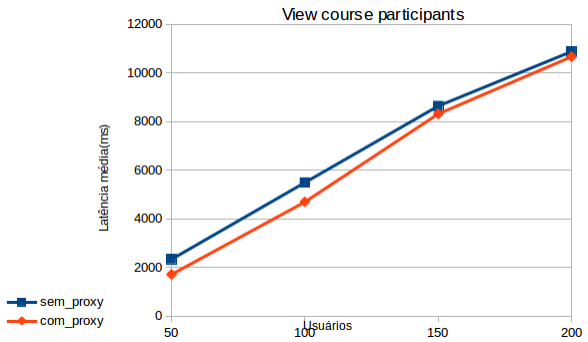

Fig. 19. Latência média da requisição Course participants de todos os usuários.

O tempo de execução dos casos de teste foi entre dois e oito minutos, que aumentou gradativamente conforme a quantidade de usuários, portanto, todos os

objetos armazenados no cache eram válidos, pois todos os objetos foram configurados para permanecer no cache por doze horas. A princípio, doze horas pode ser considerado um longo período para que um objeto fique armazenado no cache, porém, o comportamento de acesso dos alunos que estudam na modalidade a distância é consideravelmente variável, desta forma, esse período pode se tornar atraente para a economia de recursos do servidor, pois quando o primeiro usuário acessar um arquivo e este for armazenado no cache, durante as próximas doze horas todas as requisições serão respondidas pelo proxy reverso, sem a necessidade de consultas ao servidor web. Considerando a grande quantidade de alunos que a EAD atende e os inúmeros recursos didáticos disponibilizados para os alunos, a economia de recursos do servidor e melhoria do tempo de resposta pode ser expressiva.

7 Considerações finais

Este trabalho apresentou uma proposta de melhoria no ambiente virtual de aprendizagem Moodle utilizando o proxy reverso Varnish. A partir da avaliação resultados obtidos nesta proposta de integração foi possível observar uma significativa melhoria na latência das páginas que foram configuradas para serem armazenadas no proxy reverso e que as outras páginas também tiveram melhoria na letância. Tais resultados podem proporcionar um melhor desempenho nos acessos simultâneos ao servidor web do Moodle, e assim, os usuários podem ter uma latência menor ao acessar o ambiente.

Além dos benefícios que podem ser proporcionados, algumas questões devem ser consideradas. O Moodle possui um processo de auditoria que contempla todo sistema, no qual, cada ação realizada por um usuário é registrada no banco de dados [8]. Como citado anteriormente, caso um arquivo fique armazenado no cache por doze horas, um único acesso a esse arquivo deve ficar registrado no banco de dados a cada doze horas, que pode ser referente ao primeiro usuário que o acessou tal arquivo e o mesmo não foi encontrado no cache ou após o tempo de vida desse arquivo ter expirado. Em todas as solicitações seguintes a esse arquivo, o Varnish deve responder a solicitação sem a necessidade de consultar o servidor web, desde que o arquivo tenha o tempo de vida válido, portanto não haverá registro de acesso ao Moodle gravado no banco de dados, pois na prática o servidor Web não foi consultado.

Outra situação pode ocorrer quando um arquivo que estiver armazenado no cache for alterado no Moodle. Com o tempo de vida configurado para doze horas, na pior das hipóteses, os usuários podem ficar esse período de tempo acessando um arquivo desatualizado. Por fim, o servidor de proxy reverso pode ser um ponto único de falha, já que todas as solicitações destinadas ao servidor da aplicação passam antes por ele e caso ocorra alguma falha, o Moodle pode ficar indisponível.

Visto a quantidade de alunos do Ifes, dos cursos presenciais e EAD, somados a perpectiva de novos cursos e aumento da quantidade de alunos, a arquitetura apresentada pode proporcionar um melhor desempenho dos usuários no acesso ao Moodle, além de poupar recursos dos servidores.

Algumas questões não foram abordadas nesta versão da arquitetura, possibilitando alguns temas de trabalhos futuros. Dentre eles, destacamos: balanceamento de carga;

adaptar as políticas de cache para reter outros elementos do Moodle; e implementar a arquitetura proposta em um ambiente de produção.

Referências

1. Cefor.: Centro de Referência em Formação e em Educação a Distância, http://cefor.ifes.edu.br/
2. Ifes.: Instituto Federal do Espírito Santo, http://www.ifes.edu.br/
3. Moran, JM.: O que é educação a distância, www.eca.usp.br/prof/moran/site/textos/educacao_online/dist.pdf
4. Souza, GS, Leal, TACS: Educação a distância no Brasil: mudança social e tecnológica, http://www.administradores.com.br/artigos/economia-e-financas/educacao-a-distancia-no-brasil-mudanca-social-e-tecnologica/45755
5. Vilaça, M.: O que é um Ambiente Virtual de Aprendizagem (AVA)?, http://ensinoatual.com/blog/?p=137
6. TelEduc.: Bem-Vindo à Wiki do TelEduc, http://www.teleduc.org.br/.
7. EurekaBem-vindo à apresentação do Eureka, https://eureka.pucpr.br/apresentacao/index.html
8. Moodle.: Moodle - Open-source learning platform | Moodle.org, http://moodle.org/
9. Serra, A. et al.: Controle de Admissão e Diferenciacao de Serviços em Clusters de Servidores Web. In: Anais do XXIII Simpósio Brasileiro de Redes de Computadores, Forteleza (2005)
10. Patil, JB., Pawar, BV.: Improving Performance on WWW using Intelligent Predictive Caching for Web Proxy Servers. In: International Journal of ComputerScience Issues (IJCSI), v. 8, n. 1, (2001)
11. Calsavara, A., dos Santos, RG.: m algoritmo de substituição de objetos em cache na Internet baseado em semântica. In: Anais do XX Simpósio Brasileiro de Redes de Computadores, p. 135-150, Búzios (2001)
12. Wang, J.: survey of web caching schemes for the internet. In: ACM SIGCOMM Computer Communication Review , v. 29, n. 5, p. 36-46, New York (1999)
13. Barish, G., Obraczka, K.: A Survey of World Wide Web Caching (2010)
14. Nagaraj, SV.: eb Caching and Its Applications. The Springer International Series in Engineering and Computer Science, Vol. 772, 236p. Springer , New York (2004).
15. Coulouris, G. et al.: Sistemas Distribuídos: Conceitos e Projeto. 5 ed. Bookman, Porto Alegre (2013)
16. Varnish.: Varnish Community | Varnish makes websites fly!, http://www.varnish-cache.org/
17. Apache.: Welcome to The Apache Software Foundation!, http://www.apache.org/
18. Google Analytics.:
19. Google Analytics.: Website oficial do Google Analytics - análise da web e relatórios - Google Analytics, http:// www.google.com/analytics/