

# Implementação do Portal da UFRGS em Plone

Éverton Foscarini, Tiago D'Oliveira

<sup>1</sup>Universidade Federal do Rio Grande do Sul  
Centro de Processamento de Dados  
Rua Ramiro Barcelos, 2574 – Portão K – Porto Alegre – RS

{foscarini,toliveira}@cpd.ufrgs.br

**Resumo.** *Em 2010 foi tomada a decisão de desenvolver um novo Portal para a UFRGS utilizando o Plone. O grande desafio seria aliar a necessidade de um espaço que possibilitasse desenvolver um site moderno e dinâmico, que propiciasse uma fácil gestão de perfis de usuários e áreas específicas de publicação, e que, sobretudo, tivesse um bom desempenho.*

*A estabilidade e segurança foram decisivas na escolha do Plone, quando comparado com outros CMSs populares disponíveis no mercado. A experiência obteve sucesso através da combinação da gerência de conteúdos com uma adequação de processos, produtos e a implementação de uma arquitetura que viabilizasse o desempenho e disponibilidade necessárias para atender à comunidade interna e externa da UFRGS.*

## 1. Introdução

Ao longo do ano de 2009 foi realizada uma sólida discussão envolvendo diferentes equipes do Centro de Processamento de Dados da Universidade Federal do Rio Grande do Sul, acerca da necessidade de uma gerência de conteúdos mais simples e eficaz para a comunidade universitária. Nesse âmbito, procuramos avaliar no mercado as opções disponíveis e, dentre elas, a que mais se adequou às novas necessidades da Universidade foi o CMS Plone [Plone].

A decisão pelo Plone foi embasada na estabilidade e segurança da plataforma, comparada com outros CMSs populares disponíveis no mercado, como Joomla e Drupal. A extensibilidade da plataforma também era importante, para que fosse possível integrar com sistemas de gestão utilizados na Universidade. Esta integração obedece ao esquema de personalização próprio do Plone, que pode ser feita através da implementação de plugins chamados de *Plone Products*. Eles, por sua vez, contêm a lógica necessária para disponibilizar estruturas de dados que podem ser utilizadas para gerar páginas dinâmicas utilizando os dados institucionais.

Dentro de uma cronologia deste projeto, primeiramente disponibilizou-se espaço para hospedagem de sites da comunidade acadêmica e em 2010 foi tomada a decisão de desenvolver um novo Portal para a UFRGS utilizando o Plone. O grande desafio seria aliar a necessidade de um espaço que possibilitasse desenvolver um site moderno e dinâmico, que propiciasse uma fácil gestão de perfis de usuários e áreas específicas de publicação, e que, sobretudo, tivesse um grande desempenho.

Desta forma, procuramos adequar uma série de processos, produtos e serviços para que pudessemos atingir este objetivo utilizando o Plone como gerenciador de conteúdos para o Portal da UFRGS.

## **2. Produtos e funcionalidades integradoras**

### **2.1. Autenticação de usuários (OpenLDAP)**

A UFRGS utiliza o serviço de diretório OpenLDAP [OpenLDAP] para implementar a sua gestão de identidade, o que possibilita a autenticação integrada na maioria dos sistemas. A integração do Plone com o OpenLDAP era uma necessidade para evitar a criação proliferada de usuários que, além de dificultar a gestão, acabam se tornando um ponto vulnerável do sistema.

O processo de integração da autenticação em OpenLDAP com o Plone foi possível com a utilização do Produto PloneLDAP e com a adequada configuração. Isto permite utilizar as funcionalidades de Usuários e Grupos do Plone para atribuir perfis aos usuários do OpenLDAP possibilitando uma maneira simples e integrada para delegar áreas do site de acordo com a necessidade.

### **2.2. Webservices que buscam dados do Banco de Dados Institucional**

A busca de dados no Banco de Dados Institucional foi implementada no produto PortalUFRGS, que consome os dados providos por um *Webservice* disponibilizado pelo sistema de gestão da UFRGS.

Foi implementada no Produto PortalUFRGS a funcionalidade de busca de dados provenientes do Banco de Dados Institucional da UFRGS, os quais são disponibilizados através de um *Webservice*, cujo provedor está em PHP.

Para desenvolver esta funcionalidade foi utilizada a API python-suds que consome o *Webservice* via protocolo SOAP. Essa abordagem foi utilizada para algumas áreas do Portal da UFRGS, que percebeu-se ser de grande utilidade que tivessem conteúdo dinâmico. Estas áreas são as seguintes:

- Cursos de Graduação
- Cursos de Pós-Graduação (Especialização, Mestrado, Doutorado)
- Docentes
- Corpo Técnico-Administrativo

Os XMLs gerados através do *webservice* contém os dados citados acima, e cada objeto pode chegar a ter 1.2MB. Esses objetos são então associados às páginas HTML como elementos XSLT.

### **2.3. Sistema de Cache em Memória (Memcached)**

O volume dos dados gerados pelo *Webservice* é consideravelmente grande e sua geração é complexa, de forma que o acesso repetido poderia gerar atrasos na renderização de páginas pelo Plone e uma negação de serviço no provedor de *Webservice* (e conseqüentemente ao Banco de Dados Institucional) se fosse executada uma consulta a cada pageview da página do site Plone.

Para minimizar esses efeitos indesejados, utilizamos uma camada de cache intermediária para os dados retornados pelo *Webservice*, através do software *Memcached* [Memcached].

O Produto PortalUFRGS, responsável pelo acesso ao *Webservice*, armazena o resultado no serviço *Memcached* na forma de um array associativo chave-valor. A cada

página renderizada que precisa dos dados do *Webservice*, o Produto verifica se a existe a chave e se o objeto (valor) no cache ainda está válido, e se estiver utiliza-o, poupando acessos ao provedor do *Webservice* e reduzindo a latência a praticamente zero.

### 3. Infraestrutura

#### 3.1. Arquitetura dos servidores do Portal da UFRGS

A implementação da infraestrutura do Portal da UFRGS utilizando o Plone foi planejada para minimizar os pontos únicos de falha, buscando aumentar a disponibilidade do site em caso de falha e permitindo a manutenção de boa parte dos servidores sem necessidade de parada do sistema. A figura 1 apresenta o diagrama lógico dessa arquitetura.

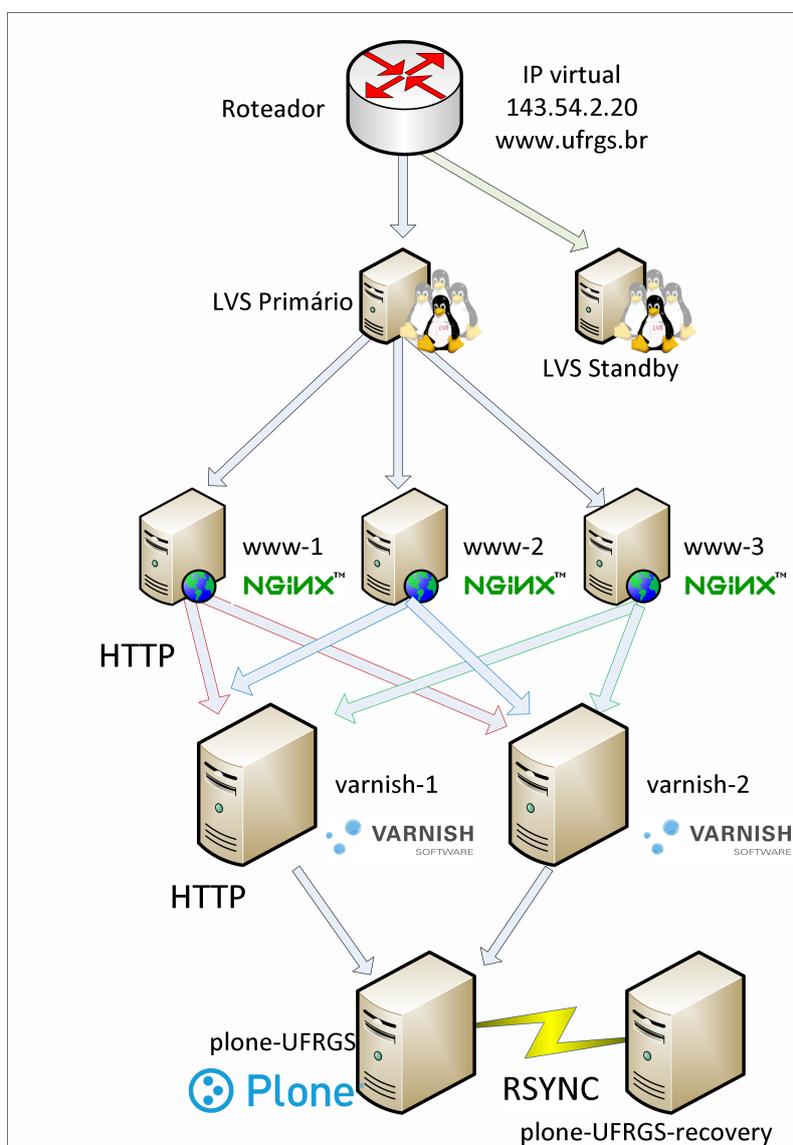


Figura 1. Arquitetura de serviços e servidores

O serviço *plone-UFRGS* roda em um servidor Ubuntu 10.04 LTS provisionado especificamente para essa função, que roda uma instância do Plone 4.1. A máquina virtual tem 6 cores Xeon E5430 e 3.5 GB de RAM. A instalação do Plone é

composta por 1 *zeoserver* (banco de dados) e 2 *clients* (motor de renderização). Periodicamente é feita uma cópia do conteúdo e das configurações para o servidor espelho chamado de plone-UFRGS-recovery, que pode ser acionado manualmente em caso de indisponibilidade prolongada do servidor principal.

O serviço *varnish* é provido por 2 servidores Ubuntu 10.04 LTS, cada um com 4 cores Xeon E5430 e 1.5 GB de RAM. É utilizada a versão 3 do varnish, e é feita a replicação de configuração entre as máquinas.

O serviço *www* é provido por 3 servidores Ubuntu 10.04 LTS, cada um com 4 cores Xeon E5430 e 512 MB de RAM. O servidor HTTP utilizado é o *nginx* [NGINX] versão 1.2, e o objetivo dessa camada é prover proxy reverso HTTP, de forma a unificar no endereço *www.ufrgs.br* para a hospedagem de sites em servidores Plone, Apache/PHP e IIS.

O serviço *Linux Virtual Server* [LVS] é um balanceador de carga de nível 4 (transporte). Para esse serviço são utilizados 2 servidores CentOS 6, configurados em modo de failover automático. Eles utilizam a funcionalidade de LVS do Kernel Linux, configurado a partir do *ldirectord*. O objetivo desse serviço é disponibilizar um IP virtual (VIP) que pode ser distribuído em múltiplos servidores, provendo balanceamento de carga e redundância.

### 3.2. Acelerador HTTP para sites Web (Varnish)

O desempenho do Plone como uma ferramenta de geração de páginas é relativamente bom se forem consideradas as possibilidades de personalização da ferramenta, porém o seu uso em um site de com grande volume de acessos é inviável devido ao custo de processamento necessário para cada página.

A seguir apresentamos alguns dados bastante significativos para o comparativo entre o uso do Plone como *webserver*, e a utilização de um acelerador HTTP para efetuar o caching dos conteúdos.

A Tabela 1 apresenta o número de requisições atendidas por segundo, e o tempo de processamento de cada um dos tipos de conteúdo listados abaixo, de acordo com o número de requisições concorrentes.

Devido à natureza monotarefa do Plone, não há diferença notável de desempenho ao paralelizar o processamento das páginas em uma mesma instância, de forma que o acesso concorrente de centenas/milhares de usuários ao site gerará uma lentidão generalizada devido à falta de *throughput*.

Para solucionar esse problema, foi utilizada uma camada de cache das páginas e dos arquivos gerados pelo Plone utilizando do software Varnish [Varnish]. Todas as requisições de clientes passam obrigatoriamente pelo Varnish, que disponibiliza imediatamente a versão da página cacheada caso ela ainda esteja no prazo de validade. É possível configurar diferentes políticas de caching, de acordo com o tipo de conteúdo, URL acessada, entre outros.

A Tabela 2 apresenta o tempo de acesso aos mesmos objetos da Tabela 1, quando o acesso é feito ao Varnish.

As medições apresentadas neste artigo foram feitas utilizando o software Apache

Tipo de conteúdo	1 conexão	10 conexões concorrentes	100 conexões concorrentes
Notícia - 18 KB (formato nativo Plone)	7 req/s 134 ms/req	8,28 req/s 120 ms/req	8,31 req/s 120 ms/req
Página Inicial - 43 K (XSLT personalizado)	4,93 req/s 202 ms/req	5 req/s 200 ms/req	4,95 req/s 202 ms/req
Página Aluno - 43 K (XSLT personalizado)	6,94 req/s 144 ms/req	7,18 req/s 139 ms/req	6,89 req/s 145 ms/req
Listagem do Corpo Docente - 3.5 MB (Dinâmico - Produto UFRGS)	0,13 req/s 7.600 ms/req	0,23 req/s 4.200 ms/req	0,25 req/s 4.000 ms/req
Arquivo de imagem - 30 KB	90 req/s 10 ms/req	130 req/s 7 ms/req	130 req/s 7 ms/req
Arquivo CSS - 1 KB	70 req/s 14 ms/req	80 req/s 12 ms/req	80 req/s 12 ms/req

**Tabela 1. Requisições por segundo e tempo por requisição - Plone**

Tipo de conteúdo	1 conexão	10 conexões concorrentes	100 conexões concorrentes
Notícia - 18 KB (formato nativo Plone)	600 req/s 1,8 ms/req	3.000 req/s 0,3 ms/req	3.500 req/s 0,3 ms/req
Página Inicial - 43 K (XSLT personalizado)	300 req/s 3 ms/req	1.200 req/s 0,9 ms/req	1.500 req/s 0,6 ms/req
Página Aluno - 43 K (XSLT personalizado)	500 req/s 2,5 ms/req	2.300 req/s 0,4 ms/req	2.300 req/s 0,4 ms/req
Listagem do Corpo Docente - 3.5 MB (Dinâmico - Produto UFRGS)	22 req/s 45 ms/req	30 req/s 35 ms/req	32 req/s 30 ms/req
Arquivo de imagem - 30 KB	400 req/s 2,5 ms/req	2.300 req/s 0,4 ms/req	2.300 req/s 0,4 ms/req
Arquivo CSS - 1 KB	1.400 req/s 0,6 ms/req	6.000 req/s 0,1 ms/req	6.000 req/s 0,1 ms/req

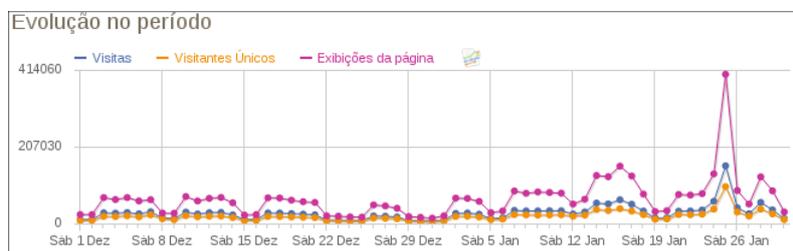
**Tabela 2. Requisições por segundo e tempo por requisição - Varnish**

Benchmark, através da execução de 100 requisições para cada tipo de página no teste do Plone, e 10.000 requisições para cada tipo no Varnish, utilizando cliente e servidor conectados através de rede Ethernet Gigabit. O servidor Plone testado é o host do Portal Plone da UFRGS em produção. O Varnish foi testado diretamente no servidor de cache, também em produção. A configuração dos servidores foi apresentada na seção 3.1. Os testes foram repetidos 5 vezes, e os valores médios foram considerados.

#### 4. Resultados

Historicamente, o Portal da UFRGS tem seu pico de acessos anual no dia de divulgação dos resultados do vestibular. O número de visitantes únicos em 25/01/2013, data da divulgação do resultado do vestibular 2013, foi de 99.273, os quais geraram 401.991 visitas a páginas apenas naquele dia. A figura 2 exibe o gráfico de acessos ao Portal da UFRGS no período de 01/12/2012 a 31/01/2013, obtido a partir do software

Piwik [Piwik ], para ilustrar a magnitude da variação do número de visitantes e de visitas no período do vestibular.



**Figura 2. Visitantes e Exibições de página**

Analisando os números a partir da perspectiva do servidor web, eles são ainda maiores. Cada página visitada pode conter dezenas de objetos, dos tipos HTML, CSS, JavaScript, imagens e outros. Analisamos os logs de acesso ao site da UFRGS e ao Plone entre 20/01/2013 e 28/01/2013, e contabilizamos o número de requisições do tipo HTTP 200 que foram executadas por cada um deles. Com base nesses valores, foi calculada a porcentagem de requisições que é tratada pelo Plone. Os resultados estão na tabela 3.

	Requisições em www.ufrgs.br	Requisições no Plone	Porcentagem que chegou ao Plone
20/01/2013	1.357.357	98.957	7.29%
21/01/2013	2.749.410	224.062	8.15%
22/01/2013	2.547.100	214.878	8.44%
23/01/2013	2.635.671	237.353	9.01%
24/01/2013	5.277.016	364.683	6.91%
25/01/2013	12.482.612	954.241	7.64%
26/01/2013	2.947.782	198.282	6.73%
27/01/2013	1.948.655	141.015	7.24%
28/01/2013	3.985.613	323.125	8.11%

**Tabela 3. Requisições por segundo e tempo por requisição - Varnish**

Os dados obtidos demonstram a eficiência do caching efetuado pelo Varnish, que atende a mais de 90% das requisições de páginas web sem repassar as requisições ao Plone.

## 5. Considerações Finais

Desde o início do planejamento para o Portal da UFRGS até a atual organização, a equipe responsável implementou uma série de modificações e adequações. A sólida percepção daquilo que se queria aliada ao aprendizado contínuo possibilitaram o constante aperfeiçoamento do produto garantindo o êxito desta experiência.

Como se pôde perceber nos resultados, conseguimos implementar as funcionalidades necessárias, tornando vantajosa a escolha pelo Plone, ao mesmo tempo que conseguimos atingir um alto desempenho, mesmo nas condições mais críticas de visitação do Portal da UFRGS.

## **Referências**

LVS. Linux Virtual Server. <http://www.linuxvirtualserver.org/>.

Memcached. Memcached: a distributed memory object caching system. <http://memcached.org/>.

NGINX. Nginx: open source web server and a reverse proxy server. <http://nginx.org/>.

OpenLDAP. OpenLDAP: a free, open source implementation of the Lightweight Directory Access Protocol (LDAP). <http://mewww.openldap.org/>.

Piwik. Piwik Web Analytics. <http://www.piwik.org>.

Plone. Plone CMS: Open Source Content Management. <http://plone.org/>.

Varnish. Varnish Cache: Web Application Accelerator. <http://www.varnish-cache.org>.