

Análise e melhoria do desempenho do serviço web do banco de termos autorizados – WSBTA do projeto Agência de Informação Embrapa.

Francisco da Silva Filho¹
Leonardo Ribeiro Queiros²

O Serviço Web do Banco de Termos Autorizados – WSBTA (QUEIROS et al., 2010) permite a manutenção de Termos Livres (termos sem ocorrência em um thesaurus) e consultas a termos do Thesaurus, Thesagro e da National Agricultural Library (NAL) pelo Sistema Gestor do Conteúdo da Agência de Informação Embrapa versão 3.0 – Gestor 3.0 (WS-BTA. Versão 1.1). É possível por esse serviço, o acesso a esses thesaurus via outros sistemas computacionais de forma automatizada, trazendo como vantagem a manutenção centralizada de termos em apenas um banco de dados. Após a primeira versão do projeto, pelos testes de desempenho, percebeu-se que o WSBTA tinha um baixo desempenho quando mais de um usuário tentava acessar simultaneamente os seus serviços. Esse baixo desempenho é um risco para sua adoção em um ambiente de produção, visto que são comuns diversos acessos simultâneos à manutenção de termos e a lentidão de acesso acarreta desconforto para os usuários das aplicações que por ventura usarão esse serviço web.

Como o problema de desempenho e falta de escalabilidade prejudica o uso do WSBTA, o objetivo principal deste trabalho é identificar e desenvolver soluções que melhorem seu desempenho para atingir

¹ Pontifícia Universidade Católica de Campinas; franciscosf@cnptia.embrapa.br

² Embrapa Informática Agropecuária; lqueiros@cnptia.embrapa.br

resultados que sejam aceitáveis a um ambiente de produção. Para viabilizar esse estudo, os principais materiais usados foram:

- Tomcat 6.0.28 como contêiner para requisições HTTP;
- PostgreSQL 8.4 que é um banco de dados open source muito bem difundido e bastante estável;
- Framework Restlet como arquitetura Restfull orientada a recursos;
- Programa desenvolvido em PHP para simular acesso de clientes ao WSBTA. Esse programa gera um número aleatório entre um e 83243 e acessa o WSBTA, recuperando o XML respectivo ao termo consultado. Esse programa ainda consegue simular N números de máquinas para acessar o WS simultaneamente;
- Sun Java 6 console para analisar o desempenho das máquinas gerando informações gráficas referentes ao consumo da memória utilizada pela JVM(Java Virtual Machine);
- Linguagem de programação Java que foi utilizada para desenvolver o WSBTA;
- Hibernate para realizar a persistência dos objetos no banco de dados relacional.

Na tentativa de resolver tal problema, a seguinte metodologia foi usada:

- 1) Trocar o contêiner Jetty pelo Tomcat 6.0.28, pois há a premissa de que o Tomcat é mais robusto para atender as requisições;
- 2) Alterar a configuração padrão de instalação do Tomcat;
- 3) Alterar a configuração padrão de instalação do PostgreSQL.

Após o WSBTA ter sido executado no Tomcat, percebeu-se um ganho significativo em relação ao desempenho. Com o Jetty, quando duas máquinas acessavam o serviço, havia um tempo médio de aproximadamente 10 segundos para atender cada requisição. Após mudar para o Tomcat, duas máquinas eram atendidas em 40 milissegundos. Além disso, existem arquivos de configurações tanto para o Tomcat, como para o PostgreSQL que ajudam a melhorar ainda mais o desempenho do projeto. Logo a seguir, as principais variáveis que foram alteradas nesses arquivos:

- CATALINA_OPTS do Tomcat. Para essa variável definiu-se os seguintes parâmetros “-server -Xms64m -Xmx128m”. Onde: -server coloca a JVM em modo servidor, que garante um melhor desempenho para atender as requisições em nível de compilação; -Xms corresponde a quantidade mínima de memória em MB que será utilizada pelo Tomcat; -Xmx corresponde a quantidade máxima de memória em MB que será utilizada pelo Tomcat;
- max_connections do PostgreSQL. Essa variável define o valor máximo de conexões suportadas pelo banco de dados e o valor padrão é 100;
- shared_buffers do PostgreSQL. Essa variável define a quantidade máxima de memória em MB que será utilizada pelo buffer do banco de dados.

Sabendo-se dos parâmetros de configuração, foram realizados diversos testes para tentar encontrar a configuração que conseguisse maximizar o desempenho do WSBTA.

No primeiro teste foi verificado que quando o programa em PHP realiza mais de 100 conexões, o PostgreSQL apresenta uma exceção (JDBC EXCEPTION) por estar configurado por padrão, para atender no máximo 100 conexões ao mesmo tempo. Assim, foi necessário aumentar o parâmetro de configuração max_connections, passando do valor padrão 100 para 200, de forma a suportar o dobro de conexões simultâneas. O segundo passo foi executar o programa PHP em duas máquinas diferentes, para verificar a influência que máquinas diferentes causam no Tomcat. De fato percebeu-se que uma máquina apenas tem um desempenho 20% melhor do que duas máquinas e quanto mais máquinas são colocadas, menor o desempenho do WSBTA.

Finalmente foram feitos os ajustes de memórias, tanto do Tomcat, como do PostgreSQL. Realizando-se diversos testes em que se variavam os valores de ambos os parâmetros, conseguiu-se maximizar o desempenho utilizando no Tomcat “-server -Xms512m -Xmx768m” e para o PostgreSQL “shared_buffers=32”.

Em vista dos resultados obtidos conclui-se que a melhoria do desempenho deve-se em grande parte na migração do Jetty para o Tomcat, e que os parâmetros de configuração ajudaram na resolução de problemas com o banco de dados e também com a grande melhora no tempo de resposta dos serviços oferecidos pelo WSBTA, que no pior teste, com 500 máquinas fazendo requisições simultâneas, o tempo de resposta chegou a ser dado em 9 segundos.

Referências

QUEIROS, L. R.; OLIVEIRA, L. H. M. de; SOUZA, M. I. F. **WS-BTA. Versão 1.1**. Campinas: Embrapa Informática Agropecuária, 2009.1 CD-ROM.