

UM ESTUDO SOBRE TIPOS DE ALGORITMOS DE DISPATCHER PARA WEB CLUSTERS

PRECIPITO, Waldemar Barilli

Docente da Faculdade de Ciências Gerenciais e Jurídicas de Garça – FAEG/Garça
waldemarbp@terra.com.br

MARANGONI, Josemar Barone

Docente da Faculdade de Ciências Gerenciais e Jurídicas de Garça – FAEG/Garça
josemarbarone@gmail.com

SANTOS, Luís Alberto Alves dos

Docente da Faculdade de Ciências Gerenciais e Jurídicas de Garça – FAEG/Garça

AOKI, Vitor

Discente do 7º Termo do Curso de Sistemas de Informação da FAEG/Garça
vitoraoki@hotmail.com

RESUMO

Este trabalho tem por objetivo apresentar um estudo inicial sobre os principais tipos de algoritmos de *dispatcher* (despachantes) aplicados a *Clusters* de *Web Services* (*Web Clusters*), sendo que são mecanismos de grande importância para o desempenho e resultados nos serviços providos pelos *Web Services*.

Palavras-chave: Algoritmos, Dispatcher, Web Service e Web Cluster.

ABSTRACT

This work has for objective present an initial study about the main kinds of algorithms of dispatcher applied to Clusters of Web Services (Web Clusters), being that are mechanisms of big importance for the performance and results in the service supplied by the Web Services.

Keywords: Algorithms, Dispatcher, Web Service and Web Cluster.

1. INTRODUÇÃO

Sistemas que necessitam se comunicar apresentam-se cada vez mais de forma freqüente. Essa necessidade de comunicação entre eles

favoreceu o desenvolvimento de tecnologias que permitem a comunicação e interoperabilidade entre eles. Uma das tecnologias que se destacam são os *Web Services*. O uso dessa nova tecnologia permite que os sistemas possam se comunicar até mesmo através da *Web*, proporcionando flexibilidade para tais sistemas, independente de tipos de plataformas e localidade das aplicações.

Devido a grande disseminação e aumento na diversidade de serviços disponibilizados, uma nova concepção de arquitetura/sistema, o *Cluster*, foi aplicada para potencializar as funcionalidades dos *Web Services*, surgindo então o conceito de *Web Cluster*.

Para que os serviços providos pelo *Web Cluster* sejam de qualidade e processados com certa velocidade, existem mecanismos que podem ser implementados em hardware ou software chamados *dispatchers*. Tais mecanismos possuem algoritmos que funcionam como o "coração" do processo, tomando decisões importantes como, por exemplo, qual o nodo do *Web Cluster* está mais disponível no momento ou qual nodo possui a melhor característica para processar determinado tipo de requisição de serviço.

Com isso, o estudo inicial sobre os tipos de algoritmos de *dispatcher* realizado nesse trabalho apresenta algumas características importantes e relevantes para o desempenho de um sistema *Web Cluster*.

2. CONSIDERAÇÕES INICIAIS

Antes de qualquer coisa, é necessário abordar alguns conceitos básicos para um melhor entendimento do próximo capítulo.

Web Services são conjuntos de aplicações autodescritivas que podem ser publicadas, localizadas e invocadas através da *Web*. Estas aplicações podem ser desde simples processos, que necessitam trocar mensagens, até complexas transações industriais, como a compra de mercadorias. Uma vez que um *Web Service* é publicado, outras aplicações

(ou outros *Web Services*) podem acessá-lo e invocá-lo, tanto para a obtenção de dados como interação com serviços que uma organização oferece (TAMAE, 2004). Diferente de outras tecnologias, *Web Services* não são acessados através de protocolos específicos de modelagem de objetos, como IIOP (*Internet Inter-ORB Protocol*), RMI (*Remote Method Invocation*) ou DCOM (*Distributed Component Object Model*), são acessados através de protocolos e formatos de dados independentes de plataforma como http (*HyperText Transfer Protocol*), XML (*Extensible Markup Language*) e SOAP (*Simple Object Access Protocol*) (GOTTSCHALK et. al, 2002).

Cluster é um sistema que compreende dois ou mais computadores ou sistemas (denominados nodos), cujo objetivo é fazer com que todo o processamento da aplicação seja distribuído aos nodos, mas de forma que pareça com que eles sejam um computador só. Com isso, é possível realizar processamentos que até então somente computadores de alto desempenho seriam capazes de fazer (AVERSA et. al, 2000). Os nodos podem ser classificados como *Front-end* (responsável pela distribuição das requisições) e *Back-end* (responsáveis pelo processamento das requisições). Portanto, um *Web Cluster* pode ser definido como um conjunto *Web Services* implementados em uma arquitetura/sistema de *Cluster*.

Os algoritmos de *dispatcher* são mecanismos que podem ser implementados em nível de hardware (mais comum) ou software que realizam o processo de análise e decisão de atribuição de requisições de serviços para os nodos do *Web Cluster*. Passa a realizar e eliminar algumas funções que o nodo *Front-end* (no sistema em *Cluster*) realizaria normalmente.

3. TIPOS DE ALGORITMOS DE DISPATCHER

Este capítulo tem por objetivos descrever sucintamente a principal característica dos tipos de algoritmos de *dispatchers* aplicados

atualmente, levando em consideração somente o fator de decisão para o encaminhamento das requisições para os nodos do *Web Cluster*.

3.1. Algoritmos *State-blind*

As políticas dos algoritmos *state-blind* não consideram qualquer informação de estado de sistema. Exemplos típicos são os algoritmos *Random* e *Round-Robin* (RR). O algoritmo *Random* encaminha as requisições recebidas uniformemente através de nodos servidores, e com probabilidades iguais de encontrar servidores. Já o algoritmo RR, realiza as decisões de *dispatching* utilizando uma lista circular com um ponteiro apontando para o último servidor selecionado.

3.2. Algoritmos *Server State-aware*

Quando são discutidos algoritmos de *dispatching* que fazem uso de alguma informação de estado de servidor, é necessário considerar o índice de carga de servidor, portanto, tomar cuidados em como e quando transmitir esta informação ao *dispatcher*.

Uma vez que um índice de carga de servidor é selecionado, o *dispatcher* pode aplicar algoritmos diferentes. Um esquema comum é designar a requisição ao servidor com o índice mais baixo de carga. O algoritmo *Least Load* (LL) indica um conjunto de políticas que dependem do índice selecionado para quantificar a carga do servidor. Existem três fatores que afetam diretamente no tempo de latência de uma requisição *Web* carregada em CPU, disco ou recursos de rede dos nodos do servidor *Web*. Típicas informações de estado do servidor incluem a utilização de CPU avaliada sobre um intervalo curto de tempo, o tamanho (comprimento) da fila processos na CPU, o disco ou taxa de E/S (Entrada/Saída) de armazenamento, o número de conexões ativas, o número de processos ativos e o tempo de latência de objeto, isto é, o tempo de conclusão de uma requisição de objeto no *Web Cluster*. Estes

índices necessitam de um monitor de processos no servidor e um mecanismo de comunicação entre os servidores e os *dispatchers*. Por exemplo, nas políticas do algoritmo *Least Connections* (LL-CONN), que são usualmente adotados em produtos comerciais da Cisco, o *dispatcher* designa as novas requisições para o servidor com menor número de conexões ativas.

3.3. Algoritmos *Client e Server State-aware*

As informações do cliente freqüentemente são combinadas com alguma informação de estado do servidor, por exemplo, fornecer a conhecida **afinidade de cliente** (HUNT et. al, 1998). Nas políticas baseadas em **afinidade de cliente**, as informações do cliente e do servidor possuem diferentes pesos: quando disponível, a informação do cliente normalmente anula a informação do servidor para decisões de atribuição. Ao invés de designar cada nova conexão a um servidor baseado em informações de estado ou atribuições anteriormente passadas, é mais vantajoso atribuir a requisição de serviço ao mesmo servidor.

4. CONCLUSÃO

Os *Web Services* surgiram com o objetivo de resolver algumas questões como integração de sistemas e facilidade na disponibilização de informações e serviços.

Com o aumento do grau de processamento das aplicações e, principalmente, o aumento nos tipos de serviços providos pelos *Web Services*, houve a necessidade de aliar disponibilidade de serviços com capacidade de disponibilizar serviços, e com isso aplicou-se a arquitetura/sistema de *Cluster* ao *Web Service*.

Os algoritmos de *dispatcher* são mecanismos de roteamento que normalmente são implementados em Hardware (*Switches*, roteadores, entre outros), com grande importância para o desempenho dos *Web Clusters*. São bastante complexos e diversificados em suas características, principalmente devido à necessidade ou não da análise de vários fatores independentes do sistema de requisição de serviços.

5. REFERÊNCIAS BIBLIOGRÁFICAS

AVERSA, L. BESTRAVOS, A. **Load balancing a cluster of web servers: using distributed packetrewriting**. Performance, Computing, and Communications Conference, 2000. IPCC '00. Conference Proceeding of the IEEE International. Fevereiro, 2000.

GOTTSCHALK, K.; GRAHAM, S.; KREGER, H.; SNELL, J. **Introduction to Web Services Architecture**. IBM Systems Journal, vol 41, No 2, 2002.

HUNT, G. S.; GOLDSZMIDT, G. D. H.; KING, R. P.; MUKHERJEE, R. **Network Dispatcher: A connection router for scalable Internet services**. Computer Networks, 30(1-7):347-357, 1998.

TAMAE, R.Y. **SISPRODIMEX – Sistema de Processamento Distribuído de Imagens Médicas com XML – Qualificação de Mestrado**. PPGCC-UNIVEM – Fundação Eurípides de Marília, São Paulo. Março de 2004.