

Controle de Admissão usando RED para um modelo Servidor Web Distribuído com Diferenciação de Serviços

Dayse Silveira de Almeida, Regina Helena Carlucci Santana

Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo (ICMC-USP) São Carlos – SP – Brasil 13560-970

{daysesa,rqs}@icmc.usp.br

Abstract. *This paper describes the experiments made with the admission control algorithm RED, implemented in a Web server model with differentiated services. We use two classes of service and a synthetic workload that varies in the arrival rate of requests to the system. It is examined that independent of workload, the class of high priority gets the best services, i.e., higher number of admissions, shorter response time, higher throughput and increased resource utilization (except when the resources are underutilized) in relation to class of lower priority.*

Resumo. *Este artigo descreve os experimentos realizados com o algoritmo de controle de admissão RED, implementado em um modelo de servidor Web com diferenciação de serviços. Utiliza-se duas classes de serviço e uma carga de trabalho sintética em que se varia a taxa de chegada de requisições ao sistema. Verifica-se que independentemente da carga utilizada, a classe de maior prioridade obtém os melhores serviços, ou seja, maior número de admissões, menor tempo de resposta, maior throughput e maior utilização dos recursos (exceto quando os recursos estão subutilizados) em relação à classe de prioridade inferior.*

1. Introdução

Com o surgimento da *World Wide Web* (WWW) na década de 90, o número de usuários da Internet cresceu fazendo com que serviço de melhor esforço (*Best Effort*) da Internet se mostra inadequado para as aplicações que, hoje, a Web comporta. Além disso, aplicações como *e-commerce* e *e-banking*, em que a necessidade de serviço possui maior exigência, concorrem de forma igualitária com aplicações convencionais da Internet, que podem ser atendidas com menor prioridade.

Para superar este serviço de melhor esforço e distinguir as aplicações que necessitam de melhores serviços, existem algumas especificações disponíveis para prover diferenciação de serviços na camada de rede, como a arquitetura de Serviços Integrados [Braden et al. 1994] e a de Serviços Diferenciados [Blake et al. 1998]. No entanto, a maioria dos servidores Web usa, atualmente, a política de atendimento de requisições baseada ordem de chegada (*First In First Out* - FIFO), o que inutiliza os esforços pela camada de rede para prover Qualidade de Serviço (*Quality of Service* - QoS).

Portanto, é necessário que a camada de aplicação também ofereça

algum tipo de diferenciação de serviços e, para isso, uma arquitetura de serviços diferenciados se mostra adequada. Assim, é necessário aplicar algoritmos que ofereçam diferentes níveis de serviços à diferentes classes de usuários.

Este trabalho avalia e analisa o comportamento do algoritmo de controle de admissão *Random Early Detection* (RED), utilizando-se um modelo de Servidor *Web* com Diferenciação de Serviços (SWDS) [Teixeira 2004], e duas classes de prioridade para o serviço.

Este trabalho está organizado da seguinte forma: na Seção 2, são apresentados alguns algoritmos de controle de admissão e modelos de servidores *Web* com serviços diferenciados. Na Seção 3 é mostrada a metodologia utilizada na avaliação de desempenho do algoritmo. Na Seção 4 é descrito o funcionamento do algoritmo RED e na Seção 5 são discutidos os resultados obtidos. Finalmente, na Seção 6, são apresentadas as conclusões e trabalhos futuros.

2. Trabalhos Relacionados

Dentre os trabalhos encontrados na literatura que desenvolveram algoritmos de controle de admissão em servidores *Web* com suporte a QoS, destacam-se os seguintes.

Em [Teixeira 2004] é proposto um modelo de servidor *Web*, composto por um módulo classificador, um módulo de controle de admissão e um cluster de servidores *Web*, que possui como objetivo prover serviços diferenciados para diferentes classes de requisições ou usuários.

Em [Chen, and Mohapatra 1999] é proposto um modelo de servidor *Web* distribuído composto por quatro componentes: um iniciador S_i responsável pelo controle de admissão; um escalonador Q , responsável pela associação das requisições admitidas a uma prioridade; N servidores de tarefas S_t , responsáveis pelo escalonamento e processamento das tarefas; e um canal de comunicação N_s , pelo qual as respostas são enviadas para os clientes.

[Serra et al. 2005] apresenta um mecanismo de para prover QoS em *clusters* de servidores *Web*, que possui três funções principais: balancear a carga imposta aos servidores, proporcionar diferenciação de serviços e, utilizar os recursos disponíveis de maneira eficaz. Para que a última função seja garantida, os recursos alocados para as diferentes classes de serviços são divididos dinamicamente, em três estados: compartilhados, exclusivos ou saturados. Quando está no estado compartilhado, o *cluster* da classe em questão possui recursos disponíveis que podem ser utilizados por outras classes. Quando passa para o modo exclusivo, significa que a carga está alta, e requisições de outras classes não são aceitas. No estado saturado, nenhuma nova requisição é aceita.

[Ye et al. 2005] sugerem um servidor *Web* composto por um servidor *Web* seqüencial com suporte a QoS, e um modelo DiffServ em que as requisições são classificadas em diferentes níveis de prioridade e colocadas em diferentes filas. Assim, as requisições de alta prioridade são sempre processadas antes das que possuem menor prioridade.

[Shan et al. 2005] apresentam um protótipo de servidor *Web* baseado em QoS, chamado QoS Apache, em que as requisições são classificadas em múltiplas categorias, e são concedidos diferentes níveis de prioridade.

Em [Messias 2007] foi desenvolvido um protótipo de servidor *Web* com diferenciação de serviços (modelo SWDS) em ambiente real, visando fornecer QoS relativa.

Com base nos trabalhos apresentados anteriormente, nota-se a busca por servidores *Web* com suporte a serviços diferenciados, a qual integra a QoS oferecida em nível de rede.

3. Metodologia

Para a avaliação de desempenho dos algoritmos, foi utilizado o modelo SWDS proposto em [Teixeira 2004]. Este Servidor *Web* com Diferenciação de Serviços foi modelado através de rede de filas e é composto por três módulos: classificador, controle de admissão e *cluster* de servidores *Web*. O classificador é responsável por dividir as requisições que chegam ao sistema em diferentes classes de serviços. O controle de admissão decide sobre a aceitação/descarte das requisições pelo sistema. E, o *cluster* de servidores *Web* processa as requisições admitidas e retornam as respostas para os clientes que as realizaram.

A parametrização utilizada considera um *cluster* com seis servidores *Web* homogêneo, em que cada servidor é modelado individualmente, com sua própria CPU, disco e interface de rede. A capacidade de processamento do classificador é definida como 8000 requisições/s e do controle de admissão como 4000 requisições/s. Para cálculo do tempo de serviço das requisições estáticas, os discos foram parametrizados com taxa de transferência de 300 MBps e o tempo de busca de 8,9 ms de acordo com as características dos discos Samsung HD161HJ SATA II [Samsung 2010], e para as requisições dinâmicas, o tempo de serviço é definido como 10 ms [Teixeira 2004].

Para realização da simulação, abordagem utilizada para validar o modelo, foi utilizado o pacote SimpackJ [Fishwick 2005]. Como carga de trabalho, foram geradas 50000 requisições para cada classe de serviço, a partir de uma ferramenta desenvolvida, similar ao HTTPPerf [HTTPPerf 2010]. O tipo de objeto requisitado é gerado de acordo com [Silva 2006], o código de *status* de acordo com [Modesto et al. 2005] e, o tamanho do objeto, de acordo com [Chehadeh et al. 2005], considerando-se os acessos as páginas iniciais e de nível 1.

Nos experimentos realizados, foram aplicados quatro tipos de carga, em que a variação da carga ocorre na taxa de requisições chegadas ao sistema por segundo. Essa taxa variou entre 300, 600, 800 e 1200 requisições/s. Os resultados obtidos correspondem a dez simulações realizadas para cada taxa, e um intervalo de confiança de 95%.

4. RED (Random Early Detection)

O algoritmo RED [Floyd and Jacobson 1993] foi proposto como forma de descartar pacotes na camada de rede e, posteriormente, foi levado para a camada de aplicação a fim de realizar controle de admissão descartando requisições em servidores *Web*.

No processamento de requisições, o RED calcula o tamanho médio da fila *avg* e o compara com um limite mínimo THMIN e um limite máximo THMAX. Se *avg* for menor que THMIN, a requisição é admitida no sistema. Se *avg* for maior que THMAX, a requisição é descartada. E ainda, se *avg* estiver entre os limites THMIN e THMAX, a probabilidade de descarte *Pa* é calculada e, então, a requisição é descartada com essa probabilidade ou admitida no sistema com a sua complementar.

O tamanho médio da fila *avg* é calculado utilizando-se uma média ponderada móvel dos tamanhos das filas anteriores. Usa-se um parâmetro fixo *wq* (peso da fila) que determina quão rápido *avg* modifica em resposta à uma alteração no tamanho atual da fila. O cálculo do *avg* é dado pela seguinte fórmula:

$$\mathbf{avg = (1 - wq) * avg + wq * q,}$$

em que, *q* é a média do tamanho atual da fila dos servidores do *cluster*.

Quando o *avg* está entre os limites THMIN e THMAX, quanto mais próximo estiver de THMAX, maior a probabilidade de descarte. Mantém-se também, uma variável contadora (*count*) para determinar o número de requisições consecutivas que escaparam do descarte; quanto maior o valor de *count*, maior a probabilidade de descarte. A fórmula da probabilidade de descarte *Pa* é dada por:

$$\mathbf{Pa = Pb / (1 - count * Pb),}$$

$$\text{com: } \mathbf{Pb = Pmax * [(avg - THMIN) / (THMAX - THMIN)],}$$

em que, *Pb* é uma probabilidade temporária usada no cálculo de *Pa* e, *Pmax* é o valor máximo que *Pb* pode atingir.

5. Resultados Experimentais

Nos experimentos realizados com o modelo SWDS e o algoritmo RED, foi utilizado um *cluster* homogêneo com seis servidores, e o algoritmo de reserva de recursos, RSV (Reserva de Recursos) [Teixeira 2004], fixando quatro servidores para a classe de maior prioridade (classe A) e dois servidores para a classe de menor prioridade (classe B). A probabilidade auxiliar *Pmax* foi fixada em 0,02, os parâmetros THMIN e THMAX foram fixados como 0,40 e 0,55, respectivamente e, o peso da fila *wq* foi fixado em 0,002.

O gráfico da Figura 1 mostra que o número de admissões ao sistema diminui à medida que a taxa de requisições/s aumenta. A admissão está relacionada à classe de serviço. Assim, a classe que possui maior prioridade (classe A) tem maior número de requisições admitidas ao sistema que a classe de prioridade inferior (classe B).

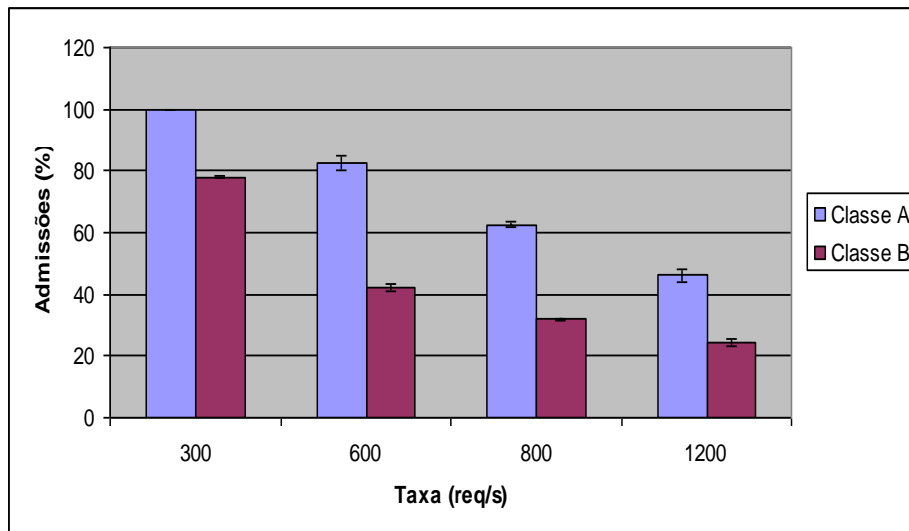


Figura 1: Admissão.

No gráfico da Figura 2 têm-se os tempos de resposta obtidos pelas classes A e B. Nota-se que os tempos de resposta obtidos pela classe A são menores que os obtidos pela classe B, pois, possui um número maior de servidores para atender as suas requisições, reservados estaticamente pelo algoritmo RSV. Assim, analisando os gráficos das Figuras 1 e 2, verifica-se o maior aproveitamento da classe A que, além de ter admitido mais requisições, submetendo uma carga maior ao sistema, obteve menores tempos de resposta para todos os tipos de carga.

Como o número de admissões diminui com aumento na intensidade da carga, os tempos de resposta não apresentam aumentos consideráveis, apenas para a classe A, de 300 requisições/s para 600 requisições/s. Isto porque, com 300 requisições/s não houve necessidade de descarte, havendo 100% de admissões ao sistema para classe A, que possui recursos suficientes para atendê-las.

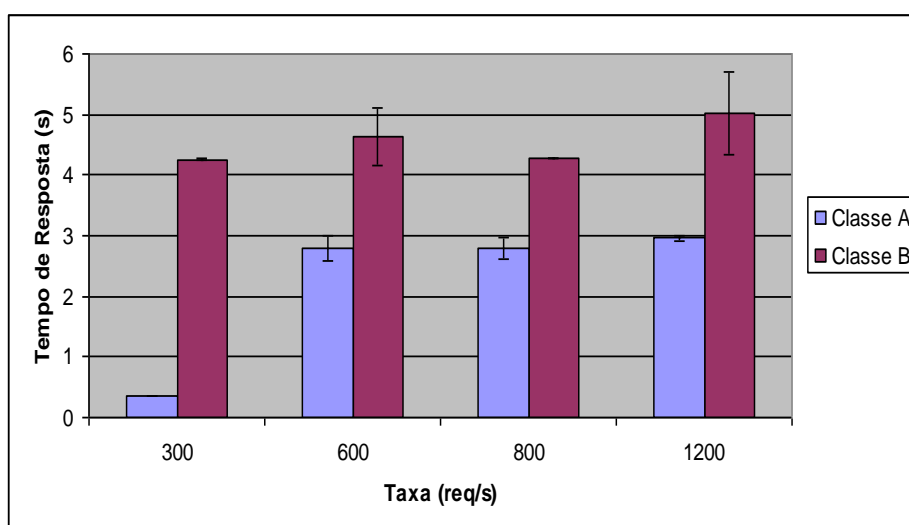


Figura 2: Tempo de resposta.

A classe A, de maior prioridade apresenta um *throughput* maior que a classe B, para todos os tipos de carga, como mostra o gráfico da Figura 3.

Assim como ocorre com o tempo de resposta, não há variação no *throughput* da classe B, e para a classe A ocorre aumento apenas quando se varia a taxa de 300 requisições/s para 600 requisições/s.

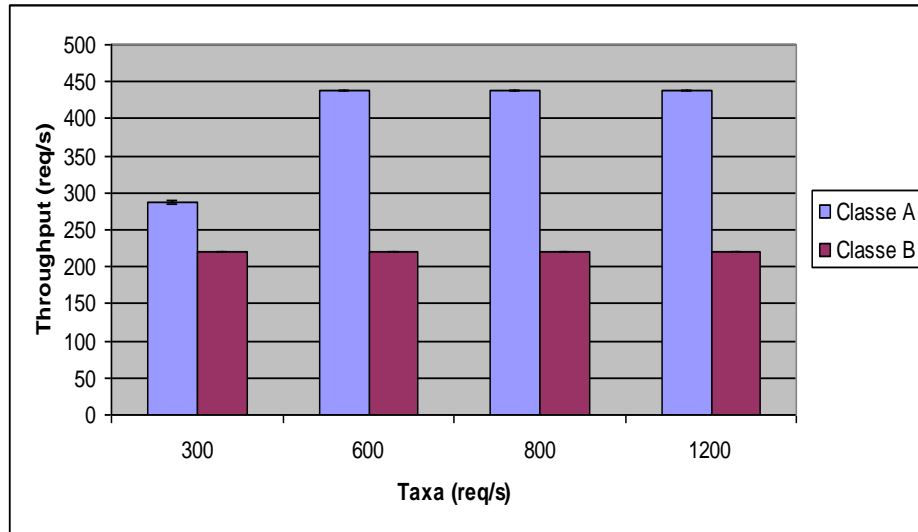


Figura 3: Throughput.

Como mostra no gráfico da Figura 4, a utilização dos recursos do sistema se mantém a mesma para as classes A e B, quando se utiliza uma carga com 300 requisições/s, pois os recursos de ambas as classes são suficientes para atender as requisições a esta taxa de chegada. Com uma taxa de chegada superior, a partir de 600 requisições/s, a classe A obtém maior utilização dos seus recursos. Observa-se um aumento na utilização dos recursos da classe A, quando se varia a taxa de 300 requisições/s para 600 requisições/s, assim como ocorre com o tempo de resposta e o *throughput*. A pequena queda na utilização dos recursos, para ambas as classes, quando se emprega uma carga com taxa de 1200 requisições/s se deve a um início de saturação, o que causa também maior número de descartes, como visto na Figura 1.

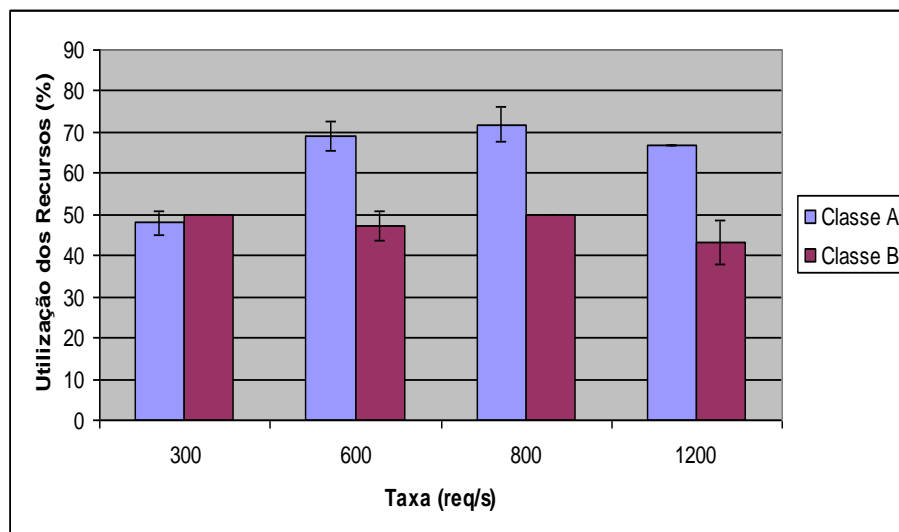


Figura 4: Utilização dos recursos do sistema.

6. Conclusões e Trabalhos Futuros

Neste artigo, foi avaliado o algoritmo de controle de admissão RED, inserido em um modelo de Servidor *Web* com Diferenciação de Serviços (SWDS). No experimento considerou-se duas classes de prioridade, a fim de se alcançar a diferenciação de serviços. E, para priorização de uma classe de serviço em relação à outra, foi utilizado um algoritmo de particionamento de recursos, o RSV, direcionando quatro servidores do *cluster* para a classe de maior prioridade, e duas para a classe inferior.

Verificou-se que a classe de maior prioridade (classe A), obteve melhor qualidade de serviço para todos os tipos de carga, tendo valores de admissão, tempo de resposta e *throughput* melhores que a classe de menor prioridade (classe B). Esta obteve também melhor utilização dos recursos, exceto quando a carga não foi suficiente para sobrecarregar o sistema.

Estes resultados mostram o comportamento do RED como controle de admissão. Como este é um algoritmo cauteloso ao descartar requisições, devido a sua região crítica, ele consegue obter uma alta utilização dos recursos do sistema.

Os resultados obtidos com este trabalho motivam a investigação do algoritmo RED com oaração com outros algoritmos de controle de admissão, como o *Drop Tail*.

7. Agradecimentos

Agradecimentos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro dado a este trabalho.

8. Referências

- Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. (1998). An Architecture for Differentiated Services. In RFC 2475. IETF.
- Braden, R., Clark, D., and Shenker, S. (1994). Integrated Services in the Internet Architecture. In RFC 1633. IETF.
- Cehadeh, Y. C., Hatahet, A. Z., Agamy, A. E., Bamakhrama, M. A., and Banawan, S. A. (2006). Investigating Distribution of Data of HTTP Traffic: An Empirical Study. *Innovations in Information Technology*, pages 1–5.
- Chen, X. and Mohapatra, P. (1999). Providing Differentiated Services from an Internet Server. In *Proceedings of the IEEE International Conference on Computer Communications and Networks*, pages 214–217.
- Fishwick, P. A. (2005). SimPackJ. University of Florida. Disponível em <http://www.cise.ufl.edu/~fishwick/simpackj/>.
- Floyd, S. and Jacobson, V. (1993). Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, pages 397–413.
- Httpperf (2010). Disponível em <http://www.hpl.hp.com/research/linux/httpperf/>.
- Silva, L. H. C. (2006). Caracterização de Carga de Trabalho para Testes de Modelos de Servidores Web. Master's thesis, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo.

- Samsung (2010). Disponível em <http://www.samsung.com/br/>
- Serra, A., Gaïti, D., Cardoso, K., Barroso, G., and Ramos, R. (2005). Controle de Admissão e Diferenciação de Serviços em Clusters de Servidores Web. XXIII Simpósio Brasileiro de Redes de Computadores (SBRC'05).
- Shan, Z., Lin, C., and Wei, Y. (2005). Prototype Implementation and Performance Evaluation of a QoS-based Web Server. Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering (SOSE'05).
- Messias, V. R. (2007). Servidor Web Distribuído com Diferenciação de Serviços - Implementação e Avaliação de um Protótipo. Master's thesis, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo.
- Modesto, M., Álvaro R. Pereira Jr, Ziviani, N., Castillo, C., and Baeza-Yates, R. (2005). Um novo retrato da Web brasileira. XXV Congresso da Sociedade Brasileira da Computação.
- Teixeira, M. A. M. (2004). Suporte a Serviços Diferenciados em Servidores Web: Modelos e Algoritmos. PhD thesis, Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo.
- Ye, N., Gel, E. S., Li, X., Farley, T., and Lai, Y.-C. (2005). Web Server QoS Models: Applying Scheduling Rules from Production Planning. *Comput. Oper. Res.*, 32(5):1147–1164.