

8 Análise de Desempenho

Conforme mencionado anteriormente, performance é um requisito importante na concepção desta arquitetura. A fim de avaliar o MoSoGw, foram aplicados alguns testes de carga utilizando o LoadStorm (38). Esses testes foram realizados em um VPS (*Virtual Private Server*) com 512MB de RAM, rodando Ubuntu 10.04. Testou-se um box MoSoGw simples com uma única instância de nginx que se comunica com uma instância de Tornado conectando-se a um serviço MySQL e memcached. Todas as três camadas de cache descritas no capítulo 4 foram implementadas.

O cenário escolhido possui os fluxos que acredita-se que serão os mais utilizados pelos usuários das aplicações sociais que utilizarem o *framework* MoSoGw:

1. Criar um novo usuário
2. Efetuar login
3. Enviar um convite para algum usuário
4. Enviar contextos de posição e mensagem de *status* de tráfego para o Twitter e Facebook através do Beanstalkd

No primeiro passo, a aplicação recebe os parâmetros enviados, acessa o banco de dados para criar o usuário e envia a confirmação da senha por email. O segundo passo consiste em obter o usuário no banco, comparar a senha informada com a salva e adicionar a sessão do usuário no memcached. O terceiro passo simula o envio de um convite para um outro usuário do sistema, havendo acesso ao banco de dados para escrita do convite na tabela correspondente. Por fim, o último passo simula o envio de dados de contexto para o Mobile Social Gateway. Esses dados são salvos no banco de dados e enviados para a fila do Beanstalkd.

O teste se inicia com 5 usuários simultâneos e, então, são adicionados novos usuários de forma linear até atingir 500 usuários simultâneos em um total de 20 minutos de testes. Os resultados são exibidos nas figuras 8.1 e 8.2. O eixo horizontal representa o tempo, em minutos, de teste decorrido e os verticais

representam o throughput (vazão, em Kb/s), na esquerda, e a quantidade de requisições por segundo, na direita. A quantidade total de dados transferidos foi de 7MB e a quantidade média de cada request foi de 0,48KB.

Na figura 8.1, pode-se observar que no momento de pico (22 req/s), o MoSoGw respondeu bem e o throughput foi proporcional ao crescimento da quantidade de usuários. Resultados alcançados são exibidos na tabela 8.1 e na figura 8.1.

Tabela 8.1: Análise de Desempenho

Resposta (média)	Requests (req)	req/s (média)	req/s (pico)	Throughput (média)	Throughput (pico)
0.177	14.810	12	22	6 kB/s	10 kB/s

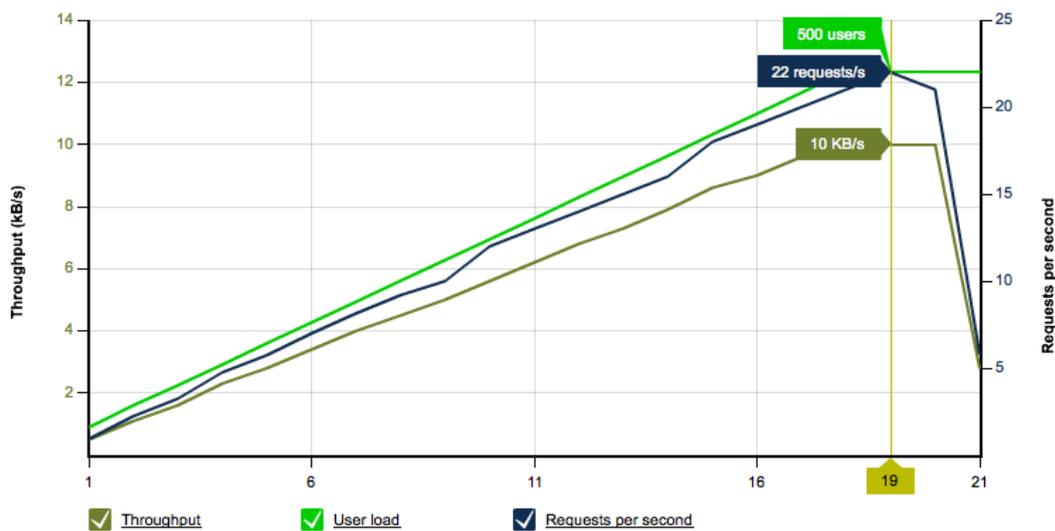


Figura 8.1: Throughput

O MoSoGw foi projetado para interagir com redes sociais, que são conhecidas por seu caráter dinâmico. Portanto, o tempo de resposta é um fator importante a ser considerado e também foi testado (figura 8.2).

Como já visto na tabela 8.1, o tempo médio de resposta foi de 0,177 segundo. No entanto, a figura 8.2 mostra um pico de 4,45s que não deve ser analisado sem um contexto. Dependendo da aplicação, mesmo dois segundos poderiam ser inaceitáveis, como no caso de aplicações para comunicação ou colaboração síncrona. Mas para o MoSoGw esse tempo significa que, em uma situação de muitos usuários simultâneos, eles podem esperar até 4,45s para obter uma resposta. O Mobile Social Gateway se propõe a enviar os dados para as redes sociais sem a necessidade de uma resposta tão rápida e o tempo médio é bastante satisfatório principalmente levando-se em consideração a configuração do servidor de testes.

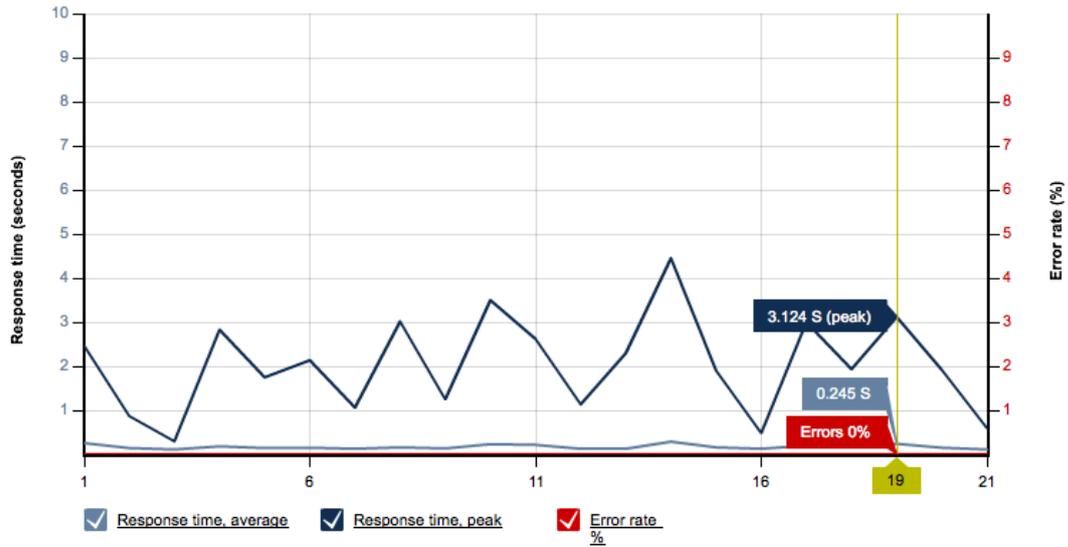


Figura 8.2: Tempo de Resposta

A curva *error rate* representa a quantidade de erros na requisições realizadas, ou seja, de requisições que retornaram uma resposta com código HTTP diferente de 200. Pela figura, pode-se observar que não houve erros durante toda a duração dos testes. Além disso, a queda de rendimento após 20 minutos não deve ser levado em consideração pois coincide com o momento em que o LoadStorm conclui a sessão de teste.