

ANÁLISE DE APLICAÇÕES DISTRIBUÍDAS EXECUTADAS EM REDES DEFINIDAS POR SOFTWARE

Gustavo Diel¹, Guilherme Piegas Koslovski²

¹ Acadêmico(a) do Curso de Ciências da Computação CCT bolsista PIBIC/CNPq

² Orientador, Departamento de Ciências da Computação, CCT – guilherme.koslovski@udesc.br.

Palavras-chave: Rede, Desempenho, Topologia, SDN, Floodlight.

As redes definidas por *software* (da sigla SDN, em inglês) possibilitam que os controladores instalados na rede tenham conhecimento total e atualizado da topologia em questão. O controlador SDN é responsável pela execução de algoritmos para encaminhamento de fluxos, que podem ser modificados de acordo com a necessidade das aplicações. Um exemplo prático de SDN, é visto em um cenário acadêmico, no qual alunos acessam uma rede sem fio. Um controlador SDN pode identificar todos os dispositivos, diferenciando quais dispositivos estão utilizando a rede para fins acadêmicos, e conseqüentemente, implementando regras de encaminhamento com prioridade para tais dispositivos. Essa diferenciação melhora o desempenho de navegação, incentivando os alunos a utilizarem a rede Wifi para estudar.

A fim de medir o quão impactante esses algoritmos podem ser quando colocados em prática, foram realizados testes de desempenho. Estes testes foram executados dentro do ambiente Mininet, um sistema que permite a virtualização em nível de sistema operacional, permitindo alocar diversos nós virtuais em um único computador hospedeiro. A partir desses nós, é montado uma topologia virtual totalmente funcional. As topologias podem ser criadas a partir de arquivos *script* em Python, e são executadas no Mininet, de forma que o mesmo se encarrega de configurar todo o ambiente de rede e de configurações.

Estas topologias foram conectadas no controlador Floodlight, um controlador SDN *open-source*, desenvolvido em Java, e que permite a criação de algoritmos especializados. Em nossos testes, utilizamos uma topologia *Fat-tree*, que pode ser visualizada na Figura 1. Os círculos indicam *switches* e os quadrados indicam nós (servidores). Foram utilizadas duas máquinas reais para simular os hospedeiros e os *switches* do Mininet. Três algoritmos de encaminhamento foram testados: Caminho mais curto (CC) que é o padrão do Floodlight; Round Robin sobre múltiplos caminhos (RR MC); e caminho aleatório (SA). Os dois últimos foram implementados no Floodlight. Cada algoritmo foi testado sob as mesmas condições, na mesma topologia, e com os mesmos testes (aplicações hospedadas).

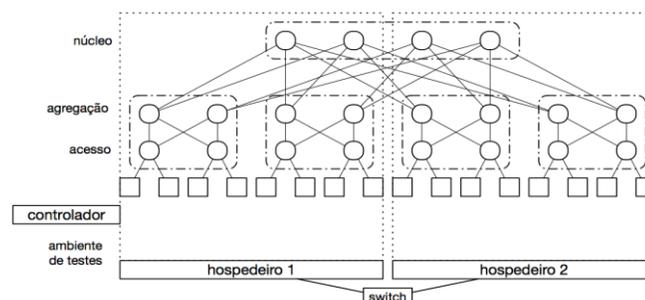


Figura 1: Exemplo de uma topologia de rede fat-tree.

Os aplicativos que compõem o *benchmark* NAS foram utilizados como aplicações alvo. Um total de seis programas da suíte NAS foi selecionado, cada um executado 10 vezes, e seus resultados estão resumidos na Figura 2 (intervalo de confiança de 95%). O algoritmo de pior desempenho é o de caminho mais curto, uma vez que este sempre seleciona a mesma rota, concentrando o tráfego, e conseqüentemente aumentando o gargalo da rede. O algoritmo RR MC obteve o melhor desempenho por sempre alternar a rota para cada fluxo, realizado um balanceamento de tráfego na topologia. Com esses resultados, é observado que diferentes regras de encaminhamento impactam significativamente no desempenho da rede.

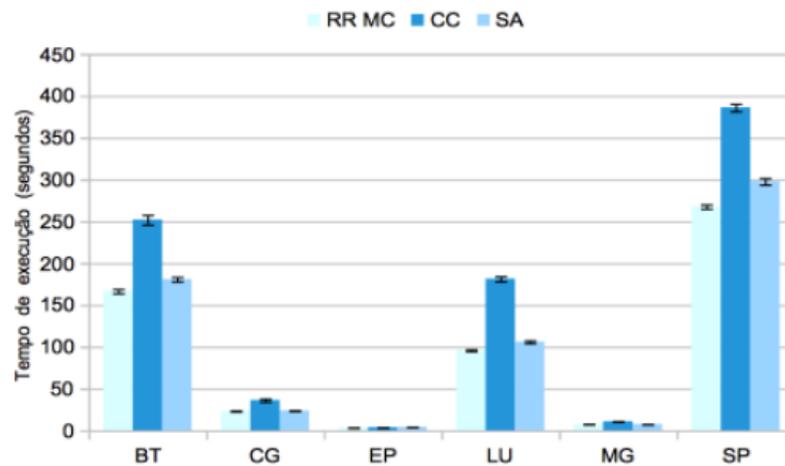


Figura 2: Execução das aplicações do benchmark NAS em uma topologia gerenciada SDN.