PROVISIONAMENTO DE REDES VIRTUAIS COM FOCO NO ESCALONAMENTO DINÂMICO DE FLUXOS

Enzo Belló Boscatto, Anderson Marcondes, Guilherme Piêgas Koslovski

INTRODUÇÃO

O desempenho de aplicações distribuídas para processamento de dados, como Apache Spark, em ambientes de nuvem e *data centers* locais, é extremamente dependente da eficiência da rede. Tráfego intenso e concorrente, especialmente durante operações de comunicação, pode levar à formação de longas filas de pacotes nos *switches* de rede, resultando em alta latência e degradação do desempenho. Neste contexto, este trabalho tem como objetivo provisionar um ambiente de rede virtual controlado para simular e analisar o comportamento do protocolo de transporte DCTCP em um *cluster* Spark, focando em como configurações e modificações na rede afetam a formação de congestionamento na rede e aperfeiçoam o gerenciamento de filas nos *switches*.

DESENVOLVIMENTO

Para a realização do estudo, foi provisionado um ambiente de teste utilizando o Oracle VirtualBox em uma máquina hospedeira utilizando Windows 11. A topologia de rede foi modelada entorno de uma rede virtualizada, consistindo em: uma máquina virtual (MV) central atuando como roteador; um *cluster* Apache Spark composto por 5 MVs; e uma MV dedicada à geração de tráfego concorrente, todas utilizando Ubuntu Server como sistema operacional. Cada MV foi isolada em sua própria sub-rede e rede interna virtual, forçando toda a comunicação entres as MVs a passar pelo nó roteador central.

Na MV roteadora, as interfaces de rede internas foram configuradas com *Random Early Detection* (RED), simulando o comportamento de um *switch* de *data center* moderno. Nas MVs do *cluster* Spark, o algoritmo de controle de congestionamento TCP foi alterado para DCTCP e o suporte a ECN foi ativado. Para a coleta de dados, foi desenvolvido um *script* em Bash que, executado na roteadora, registra em tempo real o consumo da fila em bytes e pacotes, salvando os dados em formato CSV.

A carga de trabalho foi gerada por uma aplicação PySpark customizada, projetada para induzir operações de *shuffle* durante a execução, forçando assim uma maior utilização da rede. Foram executados e comparados dois cenários experimentais para avaliar o impacto da janela inicial de congestionamento (*initcwnd*) e janela inicial de recepção (*initrwnd*) no desempenho da rede. O primeiro cenário utilizou a configuração padrão do kernel Linux com valor base 10, enquanto no segundo foi aplicada uma otimização agressiva na interface da máquina master, desligando o *Slow Start* do TCP e aumentando o *initcwnd* para 850 e o *initrwnd* para 1000, valores oriundos do cálculo de do produto entre vazão e atraso na rede.

RESULTADOS

A coleta de dados realizada na MV roteadora permitiu a observação detalhada do comportamento das filas durante todas as etapas do processo. A análise dos logs revelou que, sob carga, o tamanho da fila (*backlog*) na interface do *master* mantem-se constantemente utilizada, e durante etapas de maior utilização da rede, a fila atinge valores próximos ao limite máximo configurado. Contudo, as diferentes configurações da janela inicial de congestionamento tiveram impactos significantes nos resultados. Na configuração padrão do linux, a fila atinge maior número de pacotes apenas no estágio de maior demanda do Spark,

entretanto, no cenário otimizado a formação de fila foi drasticamente mais rápida e mais bem aproveitada durante toda a execução.

Os resultados demonstraram que os ajustes realizados diminuem o impacto do TCP *Slow Start*, permitindo que os fluxos de dados do Spark atinjam maior vazão em menor tempo, reduzindo o tempo de execução da aplicação em até 10% quando comparado ao cenário padrão.

CONSIDERAÇÕES FINAIS

Este trabalho atestou a eficácia do ambiente virtualizado criado para os testes, e todos os instrumentos utilizados para captura e análise dos dados coletados, permitindo progressão e maior qualidade no projeto. Os resultados demonstraram que o DCTCP, em conjunto com as corretas configurações de RED/ECN, gerencia ativamente o congestionamento gerado pelo *shuffle* do Apache Spark, mantendo as filas estáveis, prevenindo perdas de pacotes, e entregando resultados superiores a outros algoritmos de congestionamento. Adicionalmente, a otimização de parâmetros como o *initcwnd* acelera significativamente a utilização da banda disponível para fluxos de dados, uma otimização de alto impacto nos estágios com tarefas de menor tamanho e maior saturação da rede.

Palavras-chave: Apache Spark; controle de congestionamento; DCTCP; rede; fila; tráfego;

DADOS CADASTRAIS

BOLSISTA: Enzo Belló Boscatto
MODALIDADE DE BOLSA: PIBIC/CNPq (IC)
VIGÊNCIA: 10/2024 a 08/2025 – Total: 11 meses
ORIENTADOR(A): Guilherme Piegas Koslovski
CENTRO DE ENSINO: CCT
DEPARTAMENTO: Departamento de Ciência da Computação
ÁREAS DE CONHECIMENTO: Ciências Exatas e da Terra / Ciência da Computação
TÍTULO DO PROJETO DE PESQUISA: Mecanismos para Alocação de Infraestruturas Virtuais baseados em Aprendizado de Máquina e Acelerados por GPUs Parte 3
Nº PROTOCOLO DO PROJETO DE PESQUISA: NPP3275-2023