Podcast

Disciplina: Linguagens de programação para ciência de dados
Título do tema: Processando Big Data com Apache Spark
Autoria: Yuri Vasconcelos de Almeida Sá
Leitura crítica: Henrique Salustiano Silva

Abertura:

Olá! No podcast de hoje vamos falar sobre o dilema que o *Big data* com Spark traz e como podemos mitigar o problema de decidir a melhor infraestrutura.

A grande vantagem do apache spark sobre outros frameworks é processar os dados de forma extremamente rápida, pois o faz em memória em oposição a guardar em disco e, só aí processar os dados.

Porém o grande problema é justamente este, memória é um item caro e em grande demanda na indústria, e soluções utilizando o framework de forma distribuída e em produção pode ficar bem caro.

Basta ver o quanto de memória que as soluções prontas e máquinas virtuais que rodam o spark consomem. É uma proporção muito alta em relação aos outros frameworks.

Antes de dimensionar o seu deploy, talvez seja interessante avaliar se sua aplicação precisa de dados em streaming, que é uma das enormes vantagens que o Spark traz. Caso você não vá utilizar os dados em tempo real, talvez você possa utilizar um framework utilizando outro paradigma como o Hadoop, que usa MapReduce.

O Hadoop utiliza uma outra estrutura de processamento que envolve transferir os dados para o disco e realizar o processamento a partir de lá, que pode causar uma diferença de performance. Mas é justamente isso que pode tornar seu projeto menos intensivo, logo, menos caro.

Este é somente um dos aspectos que deve ser avaliado, a criticidade do tempo de utilização dos dados. Outros itens que podemos avaliar é o tamanho de *dataset* de produção para a devida execução do modelo.

Temos sempre que lembrar que nem todos os modelos em produção envolvem treinamento, que é a fase mais intensiva do processo. Há modelos que são executados efetuando somente o cálculo final. Muito dificilmente precisará executar ele em cluster ou grandes estruturas de processamento paralelo.

Big Data é um conceito fascinante, com diversas técnicas e estratégias que podem ser utilizadas, e os dilemas e as decisões que ele traz são tão grandes quanto os dados.

Só um exemplo simples de como fazer atualizações assíncronas no modelo, basta verificar o ciclo de vida do treinamento. Se você gera modelos que funcionam por algum tempo, podemos aguardar ou executar o treinamento em

outra estrutura enquanto o modelo de produção está rodando com dados consolidados, quando uma próxima versão estiver pronta, basta substituir o modelo em produção. Isso economiza o treinamento online, que consome bastante memória caso estejamos usando Spark.

Claro, existem aplicações que nem seriam possíveis sem um framework tão eficiente quando o Spark. Aplicações do mercado financeiro, redes sociais, controle e contenção de notícias, controle de tráfego.

Um ramo onde este tipo de processamento online em tempo real é muito utilizado é processamento de vídeo ao vivo. Processamento e reconhecimento de vídeo é amplamente utilizado no contexto Python com Spark produzindo resultados fantásticos.

Basta pensar na criticidade de reconhecimento facial em áreas de segurança, aplicações industriais de alta performance ou ainda um ramo que cresce muito, processamento de imagens médicas.

O assunto que a gente tratou aqui hoje é só um dos muitos aspectos que podemos levantar sobre o processo e estratégia do deploy e da prática em si. Mas se a gente for falar de cada etapa do processo o papo vai longe!

Fechamento:

Este foi nosso podcast de hoje! Até a próxima!