A maneira que nós mudamos a partir do padrão de errar do lado da precisão é que nós

tem maneiras diferentes de lidar com dados atrasados, marcas d'água, gatilhos, acumulações.

Então vamos falar sobre isso.

0:14

Então a ideia aqui é que existem dois conceitos.

Você tem o horário do evento, que é vinculado à origem do evento.

Por exemplo, a hora em que algo foi publicado no Pub / Sub e

então você tem o tempo de processamento.

O tempo de processamento é um risco relativo da ferramenta que processa esse evento.

Essas máquinas, os trabalhadores nos quais o DataFlow está sendo executado.

Então, entre a publicação e processamento,

lá vai ser a rede e isso vai introduzir alguma latência.

Portanto, o tempo de processamento é maior que o tempo do evento.

Deve ser maior, mas às vezes os relógios não são sincronizados e assim por diante.

Mas se os relógios estão todos sincronizados e é perfeito,

então o tempo de processamento é sempre maior que o tempo do evento.

Porque há sempre uma latência diferente de zero associada a isso.

1:12

Então, outra maneira de pensar sobre isso é que, na situação ideal,

à medida que os eventos acontecem, eles são processados ​​imediatamente.

Então, na situação ideal, o tempo do evento e

o tempo de processamento é exatamente o mesmo.

1:30

Na realidade, eventos que acontecem às 8 horas não são processados ​​às 8 horas,

1:38

eles podem ser processados ​​em 8.03.

E que os minutos da árvore são a latência, ou

outra maneira de pensar é, em 8.03,

você está processando eventos, certo, isso aconteceu às 8 horas.

Então você pode pensar nisso como um desvio de três minutos que aconteceu.

2:02

Então, isso inclina, essa diferença entre o ideal e

o tempo real do evento que você está processando, você precisa rastreá-lo,

ou, na verdade, você não precisa rastreá-lo, o DataFlow precisa rastreá-lo.

Então o DataFlow rastreia o quão longe está do horário do evento,

e é isso que a marca d'água é.

Então, a marca d'água é o DataFlow, acompanhando a distância do tempo de processamento,

ou até que ponto o tempo de processamento está no horário do evento.

2:43

Portanto, a marca d'água é uma heurística calculada a partir do fluxo.

Então, como o fluxo está chegando,

lembre-se que toda vez que algo é empurrado para o Pub / Sub.

2:56

Sub pop vai dizer, isso foi empurrado para mim, certo, às 8 horas.

E então DataFlows diz, agora é 8:03 e

portanto, parece que, com base nessa mensagem, essa fila é de três minutos.

3:13

Espero que não sejam três minutos, digamos que sejam três milissegundos.

Ok, então temos a inclinação de três milissegundos e

então olha para a próxima coisa e é uma distorção de outra coisa.

Certo, e basicamente calcula a média do desvio ao longo de um período de tempo.

Aprende esta heurística.

3:31

Mas também pode ser algo como Pub / Sub,

Você também pode definir a marca d'água como uma garantia de quão completa ela é.

Sabemos que recebemos todas as mensagens porque

Outra coisa que o DataFlow sabe é como o Pub / Sub atribuiu esses identificadores de mensagens.

Então, sabe disso, mesmo quando recebe uma mensagem em particular

tem, recebeu todas as mensagens antes deste em particular ou não.

Então, poderia ser uma heurística

poderia ser uma garantia dependendo do sistema envolvido.

Mas a marca d'água é essencialmente como esta mensagem é completa.

Então, quando entramos nessa janela embaralhada, então

você tem um fluxo ilimitado e você está fazendo um shuffle de tempo, essa é a sua janela.

E está entrando e estas são todas as mensagens entre o oitavo minuto e

o nono minuto por exemplo.

E estas são todas essas coisas que o tempo do evento é entre 8 e 9 e

aqueles entram.

Então todas as oito coisas estão na janela entre 8 e 9.

Então essa é a ideia básica.

Então, o DataFlow rastreia se a janela está completa ou não.

E a ideia da marca de seleção é completa.

E a ideia é que ainda está recebendo alguns eventos para

conheça o horário do evento entre 10 e 11.

Mas sabe que, com base em informações históricas, essa janela de tempo

É improvável que seja completo e precisa esperar antes de fechar a janela.

Então a ideia aqui é que o DataFlow está aprendendo essa marca d'água.

Esse desvio típico associado a mensagens,

quanto tempo as mensagens demoram antes de recebê-las?

Então, estou pronto para calculá-lo agora ou não é basicamente essa heurística.

O DataFlow está basicamente aprendendo como está fazendo.

5:25

Então, com o Windows, você está basicamente dizendo onde na hora do evento você deseja processar.

A marca d'água é como heurística de como eu recebi todos os dados que eu

precisa ir em frente e calculá-lo.

E um gatilho de que não falamos é quando você faz esse cálculo,

saia coletivamente,

Eles ajudam você a lidar com os dados que podem estar em ordem, que podem estar atrasados.

Então, olhamos para quatro casos diferentes aqui.

Nós olhamos como você faz isso em um lote clássico.

5:56

O que significaria fazer um lote de janela ou um lote micro.

6:00

Mas então vamos passar para a forma como você lida com streaming, e

como você lida com diferentes maneiras de fazer o trigger.

Então, bottom line novamente.

Nós conversamos sobre janelas e

janela é onde, na hora do evento, você está realmente processando.

Você está processando dados entre oito e nove?

6:21

O que você está computando, é basicamente uma transformação.

Então, estamos fazendo nossos dados de sensor, toda a extração associada a isso.

Essas são todas transformações.

E quando isso acontece no tempo de processamento?

Essa é sua marca d'água.

Porque uma marca d'água está essencialmente realizando

Por quanto tempo preciso manter a janela aberta?

Que acredito que vou recebê-lo.

E o gatilho é o que eu faço sobre dados atrasados,

porque alguma heurística está em heurística?

Pode haver dados que chegam depois que você fecha a janela.

Assim, a marca d'água ajuda o DataFlow a decidir quando é um bom momento para fechar a janela.

7:05

Portanto, o DataFlow lida com dados de streaming enquanto equilibra compensações.

O modelo de janelas permite lidar basicamente com as janelas de tempo do evento.

O modelo de disparo ajuda basicamente a vincular a saída

tempo do resultado até a maneira como o pipeline está operando.

Então você pode basicamente dizer que é assim que eu quero que aconteça o gatilho.

Vamos ver exemplos disso.

E finalmente, um modelo de processamento incremental que diz como eu volto e

recompilar meios e médias e somas, etc.

Para corrigi-los, se necessário.

E o DataFlow oferece uma implementação escalonável

de todos esses modelos. >> Quando você pretende realizar

agregações e cálculos em um fluxo de dados, você tem que decidir onde

o curso do fluxo de eventos para executar o cálculo.

Você não pode esperar até que todos os dados sejam recebidos

antes de executar o cálculo, porque o final do fluxo não é conhecido.

Pode levar dias ou pode ser infinito.

Considere, por exemplo, que você está recebendo um fluxo de números,

talvez leituras de temperatura de termômetros digitais.

Se você quisesse fornecer a temperatura média, como você faria isso?

Bem, isso depende se você está querendo fornecer a média a cada minuto,

uma vez a cada 10 minutos ou uma vez por hora.

Dividir o tempo do evento em grupos distintos de eventos é chamado de janelamento.

E com que frequência você deseja produzir resultados, determina o tamanho da janela.

Você também pode querer produzir uma média de execução.

Se o tamanho da janela é de dez minutos, então talvez em vez de um novo valor baseado

em apenas os dados que estão nos dez minutos distintos.

Você pode querer incluir cinco minutos da janela anterior e

cinco minutos da próxima janela.

Calculando a média durante 20 minutos.

Então, alguns dos valores da janela foram usados ​​duas vezes ou

várias vezes no cálculo, dando uma mudança mais suave na média.

Isso é chamado de janela deslizante.

Para dados em rajada, você pode querer ter uma janela de tamanho variável,

conectado a algum gatilho.

Por exemplo, se você estivesse medindo a precipitação média e, por alguns períodos, houvesse

sem chuva alguma, você pode querer começar uma janela ao primeiro sinal de chuva.

E na janela quando a chuva é passada.

O uso de janelas cria resultados individuais para fatias do tempo do evento.

Ele faz isso dividindo a coleção de chaves em blocos finitos baseados em

o tempo do evento de cada elemento.

O uso de janelas pode ser útil para cálculos sobre limites e

coleções de chaves ilimitadas.

Mas é necessário, ao criar agregações em dados infinitos.

9:32

Há muitas maneiras de implementar o Windows em dados.

Um método comum é definir o Windows com base em períodos de tempo fixos, como

uma hora, dia, semana ou mês.

Outro método comum é usar o Windows deslizante sobreposto.

Uma janela deslizante pode ser para

Por exemplo, as últimas 24 horas com os dados coletados a cada hora.

E, finalmente, para capturar dados em rajadas, como a atividade do usuário,

um método comum é chamado de janelas baseadas em sessão.

10:00

O código crítico neste exemplo é destacado e

Você pode ver que isso configura a janela usando janelas fixas.

Cada janela tem dois minutos, executará Sum.integersPerKey,

nos dados em cada janela.

10:15

Como esse código se comportará quando estiver em execução?

10:20

Esta animação examina a execução.

10:25

Uma resposta independente é calculada para cada dois minutos do tempo do evento.

10:31

Mas o sistema ainda está esperando até que todo o cálculo seja concluído

antes de emitir qualquer resultado.

Isso funciona bem para conjuntos de dados limitados, onde o processamento acabará por terminar.

Mas não vai funcionar quando se tenta processar um ilimitado, ou

quantidade infinita de dados.

Para produzir resultados quando estiverem prontos,

é necessário um mecanismo para controlar quando cada resultado será produzido.

10:56

Os gatilhos controlam quando o resultado deve ser emitido para uma janela.

Os gatilhos costumam ser relativos à marca d'água

que é uma heurística sobre o progresso do tempo do evento.

11:08

Acionar a marca d'água é o padrão.

É mostrado aqui explicitamente para maior clareza.

O código implementa o acionamento na marca d'água.

Observe que o acionamento diz após a marca d'água e pastEndOfWindow.

Um pedido de resultado a ser omitido será enviado quando todos os elementos para

uma determinada janela foi vista.

11:28

Esta animação ilustra a execução de um código de acionamento de marca d'água.

Existem quatro conjuntos de janelas de dois minutos,

o resultado de cada janela é omitido assim que a marca d'água passa.

O gráfico à esquerda mostra o que aconteceria com uma marca d'água perfeita.

No entanto, marcas d'água nem sempre são perfeitas.

Por exemplo,

É possível escolher uma marca d'água que depende de dados que podem ser atrasados.

Então, no exemplo à direita, um dos elementos foi atrasado.

É hora do evento diz que deve ser incluído na janela, mas

o tempo já passou.

Então esse elemento está muito atrasado.

O gráfico à direita ilustra o que acontece quando se confia em uma heurística

marca d'água.

O elemento nove estava muito atrasado devido à falta de sinal celular e

foi muito mais atrasado do que o esperado.

Então, neste caso, chegou tarde após a marca d'água e

nunca foi incluído nos resultados.

Um método mais sofisticado é necessário para lidar com essa circunstância.

Situações em que você deseja cenários de acionamento mais avançados

também exigem métodos mais sofisticados.

Por exemplo, se você quis obter resultados especulativos antes

o período da marca d'água passou em um aplicativo baseado em janelas diárias.

Embora os resultados sejam gerados,

assim que a marca d'água passar, você pode querer dados fora durante o dia.

Você pode querer obter resultados especulares iniciais.

Para que você possa ver como as coisas estão progredindo durante o dia e

sem esperar que a marca d'água passe.

Neste caso, é necessário um sistema de disparo mais avançado.

12:55

Neste exemplo, o gatilho foi atualizado para que ele não apenas emita elementos

da marca d'água, mas também cria resultados especulativos a cada minuto.

Ele continua a disparar à medida que a marca d'água passa e também quando dados atrasados ​​chegam.

13:11

Agora, a primeira janela produz um resultado na marca d'água.

Mas também fornece um resultado atualizado quando o valor final 9 é recebido.

A segunda janela tem dois resultados especulados com valores 7 e

14 antes de produzir o resultado final de 22.

Dentro de uma janela você pode descobrir quais os limites de tempo da janela

estão declarando um argumento da janela de intervalo de tipo.

Mas você não pode descobrir quantos elementos foram processados ​​na janela

porque isso é conhecido apenas pelo mestre DataFlow

que deve calcular um total em todos os trabalhadores.

13:46

Por padrão, a marca d'água é baseada no registro de data e hora da mensagem Pub / Sub.

Mas você também pode calcular uma marca d'água de ponta a ponta

definindo um timestamp personalizado com base em algum atributo da sua mensagem.

Isso pode permitir o escalonamento automático mais preciso em casos em que há atraso significativo

entre o evento original e o horário em que o Pub / Sub recebe a mensagem.