PUB/SUB

O Google fornece um serviço de comunicação muito útil para conectar sistemas e aplicativos juntos quando o tempo e o volume dos dados variam. O serviço é chamado Pub/Sub, que é um tipo de publicação-assinatura.

Publicar-assinar ou mais comumente Pub / Sub, é um barramento de mensagens que é uma ótima maneira de lidar com o desafio de ingerir volumes variáveis ​​de dados.

Pub/Sub tem a capacidade de ingerir dados em altas velocidades.

Funciona bem com dados de streaming é durável e tolerante a falhas

e porque não opera nos dados apenas armazena e encaminha,

o serviço é capaz de escalar automaticamente com a demanda.

Como você verá em breve o projeto não funciona do Pub / Sub oferece excelentes benefícios, mas também tem consequências que significam responsabilidades adicionais para seu aplicativo.

Entender as qualidades do Pub/Sub ajudará você a aprender a usá-lo. Como qualquer método de comunicação, há várias circunstâncias que devem ser consideradas, um remetente deve encontrar o destinatário para entregar a mensagem. O receptor deve estar disponível para receber a mensagem e assim por diante.

Pub / Sub lida com muitas dessas questões, liberando os comunicadores para se concentrar em seu trabalho, em vez de se concentrar no protocolo de comunicação.

O Pub / Sub lida com a descoberta do remetente e do destinatário.

Imagine se um remetente quiser se comunicar com um destinatário sem um serviço como Pub / Sub, o remetente teria que obter o endereço do destinatário. Pub / Sub age como intermediário.

O editor publica mensagens para objetos conhecidos como tópicos,

o assinante se inscreve em tópicos e recebe mensagens. O nível único de direção representado por o tópico conecta o editor e o assinante, mantendo sua independência um do outro.

O mesmo recurso de descoberta também lida com mudanças.

Se um novo assinante se unir, o editor não precisa saber sobre isso para entregar mensagens para todos os assinantes.

Se o editor mudar e um novo editor começar a publicar mensagens no tópico, os assinantes não precisam mudar.

Isso simplifica muito as comunicações. O Pub/Sub também lida com a disponibilidade do remetente e do destinatário. Ele em si é altamente disponível, se houver três editores publicando em um tópico e um vai embora, não é um problema, o tópico persistirá,

e quando o editor retorna, as comunicações são retomadas.

Da mesma forma, se um assinante é atrasado por ser sobrecarregado com outro trabalho, ou se o assinante sai de serviço, não é um problema, as mensagens são duráveis. Uma mensagem no Pub / Sub persistir por sete dias. Ele será entregue quando o assinante retornar. Assim permite o desenvolvimento de sistemas que têm qualidades disponíveis e duráveis, mesmo que seus componentes não estejam disponíveis ou sejam duráveis. O Pub / Sub é baixa latência, o que significa que você pode construa sistemas de alto desempenho e streaming usando-o, é uma infraestrutura de mensagens, você não usa para armazenar dados permanentemente, você não usa para processar ou para analisar dados, existem outros produtos para esses fins.

Pub / Sub é muito bom em capturar dados e distribuir dados, é um serviço global unificado funciona em todos os lugares da mesma maneira. não há diferentes Pub / Subs por projeto, por domínio ou por usuário, há apenas um serviço.

Não é um serviço gerenciado ele não é executado em um cluster de máquinas, é sem servidor. Uma questão comum tem a ver com a relação entre o Pub / Sub e a rede. A rede já não fornece um método para os aplicativos se comunicarem? Bem, isso é verdade, mas as qualidades da comunicação de rede básica são diferentes das qualidades que o Pub / Sub adiciona ao topo da rede.

Uma boa analogia da diferença entre a rede e Pub / Sub, é a diferença entre comunicações telefônicas e comunicações por e-mail.

O telefone é síncrono. Requer que o destinatário da chamada esteja presente no momento da comunicação e disponível ao mesmo tempo ou a mensagem não é transmitida.

E-mail, por outro lado, é um sistema de armazenamento e encaminhamento como o Pub / Sub, remetentes e destinatários não precisam estar on-line ao mesmo tempo. Ele guarda com segurança os dados até que possam ser entregues, ajuda a suavizar picos de tráfego ou comunicações em rajadas, entrega mensagens rapidamente e se auto dimensiona para lidar com volumes variáveis ​​de dados.

A coordenação de partes de um sistema usando Pub / Sub simplifica a distribuição de eventos, se as partes de um sistema precisam se comunicar diretamente umas com as outras, há muita complexidade envolvida, haveria muitos links ponto-a-ponto. No entanto, com Pub / Sub, cada parte só tem que se comunicar com um único serviço altamente disponível e que simplifica as interações.

Pub / Sub fornece comunicações assíncronas, o editor nunca espera

apenas continua publicando no Pub / Sub.  Se você tem dois assinantes e um é rápido e um é lento, não há problema com o Pub / Sub. O assinante rápido consumirá mensagens assim que forem postadas, o assinante lento recebe as mesmas mensagens, mas as consome no seu próprio ritmo.

Mesmo se o assinante ficar offline ou se tornar indisponível de alguma outra forma, a mensagem ainda será entregue por até sete dias. O Pub/Sub pode ajudá-lo a evitar o excesso de provisionamento de recursos em seu aplicativo.

Considere, por exemplo, se você teve uma situação onde são normalmente cerca de 100 máquinas necessárias. Todos os dias, durante cerca de uma hora, os picos de tráfego para lidar com o pico requerem 1.000 máquinas, usando Pub / Sub para armazenar o pedido recebido e entregá-los quando os recursos estiverem disponíveis você pode suavizar o pico de tráfego e reduzir os recursos exigidos pelo seu aplicativo. Agora você entende as qualidades do Pub / Sub.

O Pub / Sub funciona através de tópicos e assinaturas.

É um padrão de comunicação assíncrona. Pode haver vários tópicos no Pub / Sub, um ou mais editores enviam uma mensagem para um tópico. Se um assinante estiver interessado nessas mensagens podem criar uma assinatura no tópico. A assinatura é de propriedade do assinante. Se um assinante específico, como o assinante três, for embora, a assinatura é mantida por sete dias para aguardar o retorno do assinante três.

É possível ter vários assinantes compartilhando o trabalho de processar as mensagens em uma única assinatura. Por exemplo, para a assinatura 1, há assinantes um e dois, todas as mensagens do tópico aparecem na assinatura e juntos, os assinantes processarão as mensagens nessa assinatura. Esse é o método pelo qual os assinantes rápidos e os assinantes lentos podem ser misturados. Por exemplo, o assinante pode ser um parente rápido

para o assinante dois e nenhum deles tem que esperar do outro.

A dissociação de editores e assinantes garante que nenhum dos lados precisa se preocupar com a disponibilidade do outro.

É garantido que as mensagens sejam entregues pelo menos uma vez a todos os assinantes. No entanto, existem alguns comportamentos que você precisa conhecer. Não é provável, mas é possível que um assinante receba uma entrega duplicada. O que acontece é que um assinante reconhece cada mensagem para cada assinatura.

Se o assinante demorar muito para enviar a confirmação, o prazo passou e a mensagem é reenviada.

Então, se o reconhecimento de alguma forma foi descartado ou atrasado, o assinante pode obter duas cópias da mensagem.

Além disso, o assinante pode estender o prazo por mensagem para reduzir as chances de isso acontecer e também a ordem não são garantidas. Por exemplo, se você tiver mensagens de tamanhos diferentes, poderá receber mensagens menores antes de mensagens maiores em vez da ordem em que foram publicados.

Você precisa considerar como seu aplicativo lidará com

mensagens fora de ordem e possível duplicação.

Felizmente, os serviços funcionam juntos e se complementam.

Pub / Sub entrega mensagens pelo menos uma vez, mas possivelmente mais de uma vez. O Dataflow possui recursos que fornecem exatamente um processamento de tempo de uma mensagem, para lidar com duplicação.

Pub / Sub pode entregar mensagens atrasadas e fora de ordem.

O Dataflow implementa um sistema de janelas que pode manipular

dados de chegada tardia e mensagens fora de ordem.

Outra maneira de pensar sobre isso é que o Pub / Sub é assíncrono e O stateless e o dataflow têm a capacidade de tornar a solução com estado.

Pub / Sub é um serviço global. Existe um único namespace compartilhado para tópicos em todo o mundo.

Quando um editor publica uma mensagem, a mensagem é armazenada na região mais próxima do editor.

Para confiabilidade e alta disponibilidade, a mensagem é armazenada em várias zonas. Uma assinatura coleta tópicos de diferentes regiões, por exemplo, editores para o tópico a publicar mensagens. Cada mensagem é armazenada na região mais próxima do editor. A assinatura do tópico A coleta o tópico uma mensagem de todas as regiões e fornece aos seus assinantes.

Pub / Sub é rápido. A latência está na faixa de centenas de milissegundos. Também é escalável, proporcionando um desempenho consistente um kilobyte por segundo para 100 gigabytes por segundo.

O Pub / Sub suporta vários assinantes ouvindo um único tópico chamado fan-in e também suporta vários editores publicando em um único tópico chamado fan-out. O Pub / Sub também oferece push e pull para os assinantes. Em outras palavras, um assinante pode continuar verificando se uma nova mensagem está disponível, que é puxar, ou pode se registrar notificações quando há uma nova mensagem que é chamada push.

Existem também dois tipos de bibliotecas de clientes disponíveis.

Para idiomas como Java, Python, C #, Ruby e Node.js existem bibliotecas construídas manualmente.

No entanto, o Pub / Sub também suporta a chamada de procedimento remoto gRPC, que é um framework de RPC de código aberto e

bibliotecas são geradas automaticamente para idiomas gRPC.

Às vezes as pessoas olham para este diagrama e estão confusas sobre o motivo de o Pub / Sub não aparecer. Apenas para revisar, o Pub / Sub não é um banco de dados ou um serviço de armazenamento de dados, é um método de comunicação.

É usado para ingest de dados e streaming de dados para destinos de armazenamento, como um data warehouse, armazenamento em nuvem ou BigQuery. Dadas as características de baixa latência do Pub / Sub, você pode considerar usá-lo com um banco de dados rápido para aplicativos móveis. Enquanto esta é uma opção, o Firebase talvez seja uma opção melhor se o seu aplicativo envolver

comunicações pessoa a pessoa em tempo real, como para jogos,

bate-papo e streaming de atividades.