Timeout de visibilidade

Renove a visibilidade antes do vencimento

Pré-requisitos

Conheça recepção, recibo e exclusão no SQS. Use Node.js 24 para o exercício. O consumidor gera um comprovante para R45, e a duração da tarefa pode variar.

Dimensione a primeira janela [1]

A contagem começa quando a mensagem é recebida. Inclua no orçamento o tempo de espera dentro do consumidor, o processamento e a exclusão. Observe a distribuição de durações e reserve margem para latência e pausas. Uma janela longa dá mais tempo ao trabalho, mas aumenta o atraso da recuperação se o consumidor cair.

Renove a partir do instante da alteração [2]

ChangeMessageVisibility define um novo prazo a partir da chamada. Se a mensagem tinha prazo até t=30 e você define 30 segundos em t=20, o novo prazo termina em t=50. Alterar para cinco segundos no mesmo instante encurta o prazo para t=25. Use o recibo da entrega e não ultrapasse o tempo máximo restante, cujo limite é contado a partir da recepção original.

Agende com margem e acompanhe falhas

Renove enquanto há progresso, com antecedência suficiente para uma chamada lenta. Pare de renovar após concluir, desistir ou chegar ao limite operacional. Ao perder a janela, outro consumidor pode iniciar o mesmo trabalho. Preserve uma proteção durável contra duplicidade e observe erros de renovação.

Coordene a exclusão com a gravação

Depois de gravar o resultado, exclua a mensagem. Se a exclusão falhar, a próxima execução deve reconhecer o resultado. Para tarefas maiores que o prazo útil, considere checkpoints ou dividir o trabalho em etapas recuperáveis.

Timeout de visibilidade

Exercício resolvido • Calcule cada renovação

Enunciado e dados de partida

A mensagem é recebida em t=0 e fica invisível por 30 segundos. A geração do comprovante e a exclusão terminam em t=68. Simule renovações de 30 segundos em t=20, 40 e 60. Depois atrase a segunda renovação para t=50. O laboratório trata o instante do prazo como limite exclusivo e desconsidera latência.

1. Salve visibilidade.mjs

```
import assert from 'node:assert/strict';

function renovar(prazo, instante, janela) {
  if (instante >= prazo) return null;
  return instante + janela;
}

function simular(duracao, renovarEm) {
  let prazo = 30;
  for (const instante of renovarEm) {
    if (instante >= duracao) break;
    const novoPrazo = renovar(prazo, instante, 30);
    if (novoPrazo === null) return { prazo, concluiuAtempo: false };
    prazo = novoPrazo;
  }
  return { prazo, concluiuAtempo: duracao < prazo };
}

assert.deepEqual(simular(68, [20, 40, 60]), {
  prazo: 90, concluiuAtempo: true,
});
assert.deepEqual(simular(68, [20, 50, 60]), {
  prazo: 50, concluiuAtempo: false,
});
assert.equal(renovar(30, 20, 5), 25);
assert.equal(simular(30, []).concluiuAtempo, false);
console.log('Prazos de visibilidade conferidos');
```

2. Execute node visibilidade.mjs

O terminal imprime ‘Prazos de visibilidade conferidos’. A função calcula janelas e simula a falha de renovação no limite. Ela não envia requisições ao SQS.

Timeout de visibilidade

Confira a linha do tempo e os limites

3. Resolva os dois percursos

No percurso normal, os prazos são 30, 50, 70 e 90. Cada chamada precede o prazo então vigente. A exclusão em t=68 deixa 90 − 68 = 22 segundos de margem. No percurso atrasado, a renovação em t=50 encontra o prazo 50 e o consumidor já não pode depender da invisibilidade.

Caso de erro • Some ao prazo anterior

Troque o retorno de renovar por prazo + janela. No percurso normal, o simulador calcula 120 em vez de 90. A asserção falha. Na chamada com janela de cinco segundos, esse erro também impediria observar o encurtamento esperado.

Variação • O trabalho termina na janela inicial

Altere a duração para 18 segundos e mantenha a primeira renovação planejada em t=20. A rotina termina antes de renovar e usa a janela inicial até t=30. Confira a margem de 12 segundos e cancele qualquer temporizador pendente ao concluir o trabalho.

Critérios para conferir

Calcule cada janela a partir da última chamada efetiva. Confira o caso atrasado e o caso de janela encurtada. Na AWS, valide separadamente o comportamento sob latência, limite máximo e entrega repetida em uma fila de teste.

Referências primárias

[1] AWS • SQS, timeout de visibilidade e boas práticas

[2] AWS • ChangeMessageVisibility e duração da alteração

Timeout de visibilidade

Renove a visibilidade durante tarefas longas

Renove com margem

Para uma tarefa de duração variável, programe renovações antes do prazo de visibilidade e pare de renová-lo ao concluir ou abortar a execução. A renovação reduz reentregas durante o trabalho, mas o consumidor ainda precisa tolerá-las.

Exercício adicional

Uma mensagem é recebida às 10h00min00s com prazo de 30 segundos. Às 10h00min20s a chamada de renovação estabelece mais 30 segundos. Até quando vale?

Resolução

A nova janela termina às 10h00min50s, pois o prazo configurado se conta da chamada. Em um teste, simule a queda logo após a renovação e verifique que outro worker pode assumir após o novo prazo.

Ao assistir ao vídeo

Acompanhe a contagem entre o recebimento e o reaparecimento da mensagem.

Vídeo complementar

Master Amazon SQS Visibility Timeout (em inglês)