

PISID (24/25) - Síntese de Trabalho

Grupo: 9

Aluno1 : Davi de Mattos Balieiro 110916

Aluno2 : Ji Hua Zhu 110937

Aluno3 : Guilherme da Mota Castilho Caramelo Riço 111121

Aluno4 : Gonçalo Vieira Henriques 111213

Aluno5 : Rodrigo Miguel Cosme dos Santos 111255

Aluno6 : Ricardo Fernandes Paulo Isidro 111257

Breve Síntese da vossa implementação

Eficiência

Qual a componente/módulo/funcionalidade que considera que diminui mais a rapidez de transmissão de dados (desde o MQTT inicial até à tabela de medições no Mysql). (máximo 5 linhas)

De todas as vezes que corremos os Scripts e o jogo, nunca se perdeu dados desde a inserção no mongo até à inserção no MySQL.

Pensamos que o script que poderia atrasar a transmissão de dados seria o sensor.py (Mqtt->Mongo), pois é lá que verificamos os dados inválidos/anómalos/outliers e é lá que o algoritmo do “bot” joga.

Flexibilidade

A. *Qual a consequência prática caso seja adicionado ao labirinto um novo sensor de som que envia mensagens exatamente no mesmo formato (e envia no corpo da mensagem o seu ID=2). Selecionar apenas uma resposta.*

1. *Um dos programas vai “rebentar”* ____
2. *Nenhum programa “rebenta”, mas a informação apenas é registada no mongo* ____
3. *Nenhum programa “rebenta”, mas a informação apenas é registada no mongo e Mysql* ____
4. *Nenhum programa “rebenta” e a informação é utilizada para gerar alertas* __X__

B. *Qual a consequência prática caso seja necessário realojar o servidor Mysql noutra máquina (mantendo-se todo o resto). Podem escolher mais do que uma.*

1. *Tem de se parar todos os programas java que acedem ao Mysql para editar o java/python* _X_
2. *É necessário editar os scripts php* _X_
3. *Não é necessário editar nenhum programa java/python, basta alterar ficheiros de configuração e voltar a correr* ____
4. *Não é necessário editar nenhum script Php* ____

Robustez

- A. Qual a consequência prática caso o Servidor Mysql se desligue ocasionalmente durante uns segundos e automaticamente volte a ligar-se? *Selecionar apenas uma resposta.*
1. Os programas nunca param e não se pede informação__X__
 2. Os programas necessitam ser reinicializados manualmente, mas não se pede informação____
 3. Os programas nunca param, mas perde-se informação____
 4. Os programas necessitam ser reinicializados manualmente e perde-se informação____
- B. Qual a consequência prática caso dois servidores mongo se desliguem ocasionalmente durante uns segundos e automaticamente voltem a ligar-se? *Selecionar apenas uma resposta.*
1. O programa java/python nunca para e não se pede informação__X__
 2. O programa java/python necessita ser reinicializado manualmente, mas não se pede informação____
 3. O programa java/python nunca para, mas perde-se informação____
 4. O programa java/python necessita ser reinicializado manualmente e perde-se informação____

Segurança de Dados

Qual a componente/módulo/funcionalidade que considera que expõe mais os dados a intrusos? (máximo 5 linhas)

Supondo que os intrusos acedem apenas aos dados, a componente mais vulnerável é o mongo.py (manda dados do MongoDB local para um tópico de MQTT), porque qualquer um que tiver acesso ao tópico recebe esses dados, sendo que não fazemos uma autenticação nenhuma, nem fazemos uma criptografia às mensagens

Comparação MQTT versus “ligação Directa” (a que, por exemplo, têm com a leitura da bd da nuvem)

Flexibilidade

Qual a solução mais flexível em termos de, por exemplo, pretender mudar de tecnologia de bases de dados ou de servidor?

Assumimos como ligação direta, ligar do servidor que gerencia as mensagens do Mazerun diretamente para a tecnologia de bd.

Em termos de escalabilidade, o MQTT é mais adequado, pois faz o tratamento de dados e manda os dados “válidos” para todos, porém em termos de conectividade entre tecnologias de bases de dados, pensamos que são as duas iguais, porque ao mudar a tecnologia, sendo mqtt ou ligação direta, iríamos ter de mudar a sintaxe para conexão à tecnologia de base de dados em ambos, e adicionalmente, fazer o tratamento de dados.

Assumindo que o servidor é o que gerencia as mensagens vindas diretamente do Mazerun. Se o servidor mudar, é mais flexível mudar apenas o código de um computador (abordagem mqtt) e este continuar a mandar dados válidos para outros computadores do que ligação direta pois teria também de fazer validação de dados.

Robustez

Qual a solução mais robusta em termos de, por exemplo, componentes do sistema “rebentarem” (servidores de dados, programas)?

Solução da ligação direta, pois depende de menos programas, há menos passos intermediários entre o servidor e BS, por isso há menos chances de haver programas que rebentem. O MQTT envolve o broker e subscritores, logo há mais pontos de falha.

Ou seja, na ligação direta, só os servidores é que conseguem rebentar, enquanto que no mqtt, os servidores podem arrebentar mas o broker e os subscritores podem rebentar também.

Segurança de Dados

Qual a solução que mais expõe os dados a possíveis intrusos?

Solução do MQTT, pois é fácil de aceder e subscrever à nuvem MQTT, pois não é implementado nenhum sistema de segurança, podendo depois obter os dados inseridos lá. Por outro lado, na ligação direta só tem acesso aos dados na BD ou servidor quem tem as credenciais válidas.

Comparação com outro grupo

Seis principais vantagens da vossa solução face à solução “best of” do outro grupo

1

Periodicidade de 0s para ir buscar ao mongo, fica sempre em “escuta ativa” e não há atrasos para visibilidade no android/inserção no MySQL

2

A maneira como tratamos dos outliers é flexível ao correr os scripts, por omissão de parâmetro o valor é 10s. Decidimos implementar assim a decisão dos outliers para não sobrecarregar o user com escolhas de parâmetros.

3

Decidimos manter o campo “Gatilhos” na tabela sala pois assim conseguimos verificar se o número de gatilhos acionados numa certa sala já ultrapassou o número máximo de gatilhos possíveis (3), usando depois essa *feature* pelo android.

4

Assumimos que o outro grupo não fez verificação para duplicados de QOS 1 (na parte de receção) para além do campo “processed”, logo na receção pode haver duplicação de dados e haver então incoerência na base de dados MySql. Tal não acontece no nosso porque utilizamos QOS 2, que garante só envio e receção de uma mensagem.

5

Finalizamos o jogo com um trigger que verifica os últimos 30 movimentos a status=2, para que posteriormente altere o jogo para finalizado.

6

Para marcar pontos com o bot, ler da BD é mais simples, mas se calhar consome mais tempo, até porque tem de se esperar que o programa acabe de lá inserir os dados. Logo se correremos o jogo com um baixo intervalo entre movimentos, a chance de o que está na BD e o estado “atual” do jogo estarem incoerentes fez com que nós implementássemos o bot no python apenas.

Algo que considerem “errado” na solução do outro grupo?

1

Consultam a bd para verificar o equilibrio entre marsamis odd e even, o que, por ter de esperar que os dados sejam inseridos na bd, faz com que o bot esteja suscetível a delays na sua ativação de triggers e por consequência não pontue como desejado.

2 Assumimos que o outro grupo não fez verificação para duplicados de QOS 1 (na parte de recepção) para além do campo “processed”, logo na recepção pode haver duplicação de dados e haver então incoerência na base de dados MySql.