SAGeo: Simulador para Sistemas de Bases de Dados Geo-Replicadas

Paulo Sousa[1], José Pereira[1][0000−0002−3341−9217] e Ricardo Vilaça[1][0000−0002−6957−1536]

INESCTEC e Universidade do Minho, Braga, Portugal

Resumo Para fazer face ao aumento do número aplicações com acesso a grandes quantidades de informação, têm sido apresentados diferentes algoritmos de bases de dados em grande escala, normalmente separados em dois tipos: os que dão prioridade à performance e à disponibilidade dos dados e os que dão prioridade à coerência dos dados. Destes algoritmos é difícil perceber as diferentes entre eles, devido a diferentes abordagens na sua elaboração e avaliação. Neste artigo propomos o SAGeo, um modelo configurável de simulação de sistemas gestão de dados geo-replicados desenvolvido na ferramenta SimPy. Como demonstração da sua utilidade, mostramos que é capaz de reproduzir as caraterísticas do sistema Wren em termos de tempo de resposta e o débito, com e sem a utilização de cache.

Palavras chave: Geo-replicação, Simulação, Bases de dados

1 Introdução

Existe uma necessidade crescente de sistemas de gestão de bases de dados capazes de armazenar a informação de aplicações em grande escala. É desejável destes serviços um baixo tempo de resposta, débito elevado, coerência de dados e alta disponibilidade. Deste modo, têm sido apresentados diferentes algoritmos de bases de dados geo-replicadas, geralmente divididos em duas famílias de acordo com o comportamento na presença de partições da rede [**?**].

Por um lado, nos sistemas geo-replicados que privilegiam a disponibilidade, um dos primeiros sistemas a ser apresentados com estas características foi o Dynamo da Amazon [**?**], um sistema replicado e particionado com coerência eventual. Esta proposta foi melhorada com a introdução de coerência causal [**?**] e transações [**?**]. Estas caraterísticas foram também combinadas entre si [**?**] e otimizadas para permitir replicação parcial, em que nem todos os itens de dados estão presentes em todos os centros de dados [**?**,**?**]

Por outro lado, existe também uma grande variedade de propostas de sistemas que privilegiam a coerência. Os sistemas Spanner [**?**] e CockroachDB [**?**] oferecem *serializability* em múltiplos centros de dados e com replicação parcial, no primeiro caso, usando relógios sincronizados e no segundo caso recorrendo a um protocolo de consenso. Existem também alternativas que suportam isolamento mais fraco no sentido de melhorar o desempenho [**?**,**?**].

Em resumo, estes algoritmos servem propósitos diferentes e abordagens de implementação distintas. Além disso, mesmo dentro da mesma família de algoritmos, existe um vasto leque de implementações diferentes devido à constante evolução. Devido à grande variedade de abordagens é muitas vezes difícil perceber as diferenças entre eles e quantificar as vantagens e desvantagens relativas.

Para resolver este problema propomos o SAGeo,[1] um modelo configurável de simulação de bases de dados desenvolvido na ferramenta SimPy [**?**]. Este modelo foi configurado de forma a simular uma gama de diferentes algoritmos de replicação de dados, como DBSM [**?**], Wren [**?**] e Spanner [**?**]. Neste trabalho descrevemos o modelo genérico e fazemos o estudo de um caso da sua aplicação ao Wren [**?**], que pode ser utilizado para reproduzir as caraterísticas em termos de tempo de resposta e débito, com e sem a utilização de cache.

O resto deste artigo está organizado da seguinte forma: a Secção 2 descreve o modelo configurável de simulação de bases de dados geo-replicadas, a Secção 3 demonstra a utilidade do SAGeo no estudo de um caso da sua aplicação ao Wren, e por fim, a Secção 4 conclui o trabalho apresentado e apresenta o trabalho futuro.

2 Modelo genérico

O SAGeo é um modelo de simulação de bases de dados geo-replicadas desenvolvido na ferramenta SimPy.[2] Este modelo-base pode ser configurado de modo a corresponder às especificações do sistema de base de dados em estudo. A implementação de uma nova configuração é um processo bastante simples, sendo apenas necessário alterar pequenas parte do código no que toca a troca de mensagens e, caso necessário, adicionar componentes modulares, para implementar um novo algoritmo.

O modelo é composto por um conjunto de centros de dados e pela rede, que é um recurso partilhado por todos os centros de dados. Cada centro de dados é modelado por um processo e é composto por um gerador de carga, que gera transações a serem executadas na base de dados, e um número finito de fragmentos (servidores), sendo cada um responsável por uma partição dos dados. O gerador de carga é também modelado por um processo e gera transações com intervalo que segue uma distribuição de Poisson. O tamanho da transação segue também uma distribuição de Poisson e a localidade dos dados selecionados segue uma distribuição Zipfian.

Cada fragmento é modelado por um processo e é composto por um CPU (com um número de núcleos configurável) e um conjunto de discos (também configurável). Estes recursos são partilhados por um conjunto de fios de execução responsáveis por atender pedidos de clientes e um segundo conjunto de fios de execução responsáveis por atender pedidos de outros servidores.

Além desta arquitetura base é possível adicionar componentes ao modelo de modo a corresponder com as funcionalidades que estamos a simular. Para além

[1] O código fonte está disponível em `https://github.com/29medium/SAGeo`.
[2] `https://simpy.readthedocs.io/en/latest/`

disso, o código executado para atender cada pedido e a geração dos pedidos também são editáveis, de modo a poder ser ajustado ao comportamento do sistema em questão.

No nosso modelo de simulação os parâmetros são introduzidos através de um ficheiro de configuração em formato JSON, onde são indicados todos os valores de simulação. Alguns destes parâmetros são o número de servidores, o número de centros de dados, o tempo de simulação, etc. Além destes parâmetros é também possível adicionar novas variáveis, de modo a aproximar o modelo de simulação da base de dados em questão. Além dos parâmetros estáticos na simulação, é também introduzido um parâmetro que é variado na simulação, de modo a poder comparar o efeito da sua variação no comportamento do sistema.

Após a execução do modelo, é extraído de cada execução o tempo de resposta médio de pedidos, o débito e o número de transações executadas. Estas estatísticas são depois apresentadas em 4 gráficos: o parâmetro a variar em função do tempo de resposta, o parâmetro a variar em função do débito, o débito em função do tempo de resposta e o parâmetro a variar em função do número de transações.

3 Estudo de caso: Wren

O sistema Wren [**?**] propõe um algoritmo de bases de dados geo-replicadas que suporta coerência causal transacional com leituras não bloqueantes e, ao mesmo tempo, permite à aplicação escalar horizontalmente através de fragmentação (*sharding*), apresentando uma solução com baixa latência e disponível perante partições de rede. Para isso implementa um sistema distribuído de armazenamento chave-valor multi-versão com N partições, em que cada item é atribuído a uma partição através de uma função de *hash*. Os dados são totalmente replicados por todos os centros de dados de forma assíncrona. O algoritmo resolve os conflitos de escrita escolhendo o mais recente, ou seja, com a regra conhecida como *last-writer-wins*.

Para simular o sistema Wren com o SAGeo, primeiramente, o código executado por cada fio de execução no servidor é modificado, de modo a garantir que os passos do algoritmo são simulados corretamente. Além disso, é adicionado um componente cliente composto por um registo de transações efetuadas que tem o propósito de funcionar como cache. Quando o cliente recebe um pedido de leitura do gerador de carga verifica na sua cache se contém o pedido e, caso contrário, envia o pedido ao fragmento coordenador. O gerador de carga é também alterado, de modo a que cada transação tem associado um cliente e um fragmento coordenador que irá receber a transação. Por último, existem mais três processos a correr no servidor responsáveis por manter atualizados os relógios lógicos e por propagar as transações para os outros centros de dados.

Embora uma simulação não seja capaz de reproduzir um caso concreto de um sistema real pode observar-se que a variação de parâmetros como a existência de uma cache e a latência entre centros de dados conduz a variações semelhantes às observadas num sistema real.

4 Discussão

De forma a responder à necessidade de comparação em abstrato de diferentes algoritmos para sistemas de gestão de bases de dados geo-replicadas, este trabalho propõe o SAGeo, um modelo de simulação modular que pode ser adaptado a diferentes algoritmos e configurado para uma gama de ambientes. A utilidade deste modelo é avaliada com uma configuração para o sistema de bases de dados Wren [**?**], onde o modelo de simulação proposto consegue reproduzir o comportamento descrito na sua concretização original.

Normalmente, um protótipo do sistema que é proposto é testado usando um conjunto de servidores na nuvem geograficamente dispersos, frequentemente, incluindo uma comparação pontual com um outro sistema percursor. Apesar de útil para demonstrar a viabilidade de cada proposta numa situação realista, esta abordagem tem várias limitações. Em primeiro lugar, a escala a que os testes são feitos em termos do número de servidores é limitada pelo custo da infraestrutura, o que impede a avaliação de condições relevantes [**?**]. Em segundo lugar, não se dispõe de concretizações comparáveis de múltiplos sistemas que possam mostrar claramente os diferentes compromissos. Finalmente, a reproducibilidade de experiências na nuvem é difícil de atingir [**?**,**?**].

Tal como acontece com a generalidade dos sistemas de computação, existem várias propostas de utilização de simulação na avaliação de sistemas de bases de dados, mas são focadas na concorrência interna do servidor não na sua distribuição [**?**,**?**]. Uma abordagem particularmente interessante é a concretização de um sistema real mas simplificado com variantes para diferentes métodos de controlo de concorrência [**?**] que depois é executado num simulador de sistema completo [**?**] para analisar em detalhe o comportamento em plataformas com um grande número de núcleos, mas que reduz a escala dos modelos tratáveis.

A implementação e avaliação de algoritmos distribuídos em grande escala, em especial os que abordam desafios relacionados com a tolerância a faltas e a resiliência, é uma tarefa complexa, e o desempenho e o comportamento destas aplicações são altamente influenciados pelas propriedades da rede. Desta forma ferramentas como Babel [**?**] e Kollaps [**?**] ajudam na rápida e eficiente prototipagem de especificações ou algoritmos assim como a emular/simular rede e sistemas distribuídos complexos. No entanto, estas ferramentas são genéricas e os algoritmos de bases de dados geo-replicadas são complexos, mas existe uma base comum para a simulação das propriedades dos mesmos que tirámos vantagem neste artigo ao propor um modelo configurável de simulação.

Como trabalho futuro planeamos configurar o nosso modelo a outros algoritmos de bases de dados e avaliar o seu desempenho, podendo assim comparar os resultados obtidos de diversos sistemas e melhor compreender o impacto de cada uma das suas caraterísticas e diferenças.