Confiabilidade de Sistemas Distribuídos 2016/2017

Trabalho Prático de avaliação nº 2

DDS *Dependable Storage System* V2

Versão do enunciado: 2/Maio/2017

Resumo

Neste trabalho iremos iterar o protótipo do trabalho prático de avaliação nº 1 de modo a permitir a adição das seguintes funcionalidades: (i) implementação da API disponibilizada ao cliente para suporte das operações da interface estendida cuja especificação base se encontrava no enunciado do trabalho prático 1 mas que era ainda suportada pelo sistema DDS; (ii) suporte para armazenamento de dados mantidos sempre cifrados, de modo a permitir que as operações da interface (base e estendida) possam ser realizadas sobre os dados cifrados e (iii) adição de suporte estendido para tolerância e recuperação de falhas e ataques bizantinos, com garantia de disponibilidade e operação contínua do sistema.

1. Introdução

As três funcionalidades a adicionar ao sistema anteriormente descritas deverão ser concebidas de modo a serem adicionadas à implementação atual do sistema DDS resultante do trabalho prático nº 1. Assim, a concepção e desenvolvimento destas funcionalidades deve ter em conta o sistema de replicação com tolerância a falhas bizantinas anteriormente adotado, a saber:

- baseado numa implementação que tem por base o algoritmo ABD num ambiente de quóruns bizantinos;
- baseado numa implementação que tem por base um ambiente de replicação SMR baseada no algoritmo PAXOS, alicerçado na biblioteca BFTsmart.

Na implementação atual, não sendo o caso, deverá ser adicionada flexibilidade de configuração do suporte TLS, quer na proteção da interação cliente/sistema DDS, como na comunicação entre as réplicas do serviço de armazenamento replicado. Para o efeito deve ser possível configurar as *ciphersuites* (*handshake* TLS) de todos os *endpoints* envolvidos, tendo por base a configuração de ciphersuites consideradas seguras e descartando-se a possibilidade de *setup* com *ciphersuites* consideradas inseguras ou frágeis, no estado da arte atual. Verificar a recomendação:

- https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet

2. Aspetos de implementação

Para todos os aspetos a seguir indicados, deve ter-se em conta a especificação inicial que vinha do trabalho prático nº 1. As seguintes secções abordam cada uma das três funcionalidades a adicionar para o trabalho prático nº 2 (DDS v2)

2.1 Implementação da API estendida

O sistema suportará agora as operações da API inicialmente apenas definida na interface cliente/servidor na implementação inicial do sistema DDS (v1), seguindo o mesmo modelo de sistema e arquitetura anteriormente implementados.

2.2 Proteção dos dados e suporte para operações sobre os dados cifrados

Para este efeito será utilizada uma biblioteca JAVA implementada no DI-FCT-UNL (SJHomoLib – Synthetic Java Homomorphic Library) e que disponibiliza diversos algoritmos parcialmente homomórficos e respetivas construções criptográficas para suporte de várias operações, a saber:

- Encriptação para comparação determinística de valores (*CHE – Comparative Homomorphic Encryption*);
- Encriptação com preservação de relações de ordem (*OPE-Order Preserved Encryption*)
- Encriptação para suporte de pesquisas lineares sobre strings (ou LSE - Linear Serachable Encryption)
- Encriptação para soma de dados cifrados (ou *PSSE – Paillier Sum Supported Encryption*)
- Encriptação para multiplicação de dados cifrados (ou *MSE – Multiplication Supported Encryption*)

Para conhecimento de *background* sobre os algoritmos e construções criptográficas disponibilizadas deverá consultar-se o documento em [1][1].

2.3 Recuperação dinâmica com garantia de disponibilidade permanente

Para se abordar esta funcionalidade dever-se-á considerar o seguinte modelo de implementação (tendo por base os mecanismos iniciais já subjacentes ao modelo de replicação usado na implementação do trabalho prático nº 1).

2.3.1 Detecção por suspeição de falhas

Sempre que uma réplica revelar um comportamento incorreto (devido a falha por paragem ou por exibição de uma falha bizantina detetada pelo protocolo de replicação, seja esse comportamento devido a falhas acidentais ou indução maliciosa resultante de uma intrusão), esse evento deve ser gerido no sistema de modo a desencadear um processo de coordenação da detecção dessa falha. Para o efeito, será necessário que o sistema utilize os respetivos mecanismos de quórum ou de consenso de modo a que as réplicas possam acordar numa mudança de vista com descarte da réplica em causa.

Como possível aproximação à solução poder-se-á considerar um servidor que será responsável por indicar que suspeita de uma réplica quando:

(1) recebe uma evidência de comportamento bizantino ou notificação de falha comunicada por outra réplica que detectou quebra de comunicação com a réplica em causa K

(2) recebeu um quórum de mensagens das réplicas a indicar que suspeita da falha de uma réplica. No caso de implementações com o akka, esta solução poderá ser implementada por um ator que eventualmente já e responsável pelo *setup* das réplicas no sistema.

---

[1] ***Nota.*** A biblioteca HomoLib contém ainda outros algoritmos e construções criptográficas para manipulação de queries SQL sobre bases de dados cifradas (SQL-E Q Encrypted Queries) bem como para pesquisas para classificação analítica de imagens cifradas (Image-Based Serachable Encryption), mas este suporte estendido não será utilizado no trabalho.

2.3.2 Recuperação por decisão de detecção de falhas

Após a detecção, deverá ser assincronamente recuperada uma réplica fresca para substituição da réplica falhada ou maliciosa que foi descartada. Para o efeito será necessário que se proceda à instalação dessa réplica, por transferência do estado correto na visão das réplicas corretas.

2.3.3 Instalação da nova vista de réplicas corretas

Após a prontidão da nova réplica correta (em *correct steady-state*), deverá ser instalada a nova vista que a incorpora (o que implica o quórum ou consenso para o efeito), de modo que o sistema não perca a disponibilidade. Este aspeto deve ser verificado, no caso de implementações usando o sistema BFTsmart, a partir de suporte já existentes.

2.3.4 Garantia de disponibilidade permanente

Para se prevenir que nunca se verifique uma situação de quebra de condições de quórum ou de consenso para efeitos dos passos anteriores, deverá ser definido um valor *threshold* K de réplicas no sistema, em que K será adicionado ao fator de resiliência no quórum ABD ou no consenso SMR-PAXOS (BFTsmart) necessários. O mecanismo de detecção e recuperação deve iniciar-se sempre que se baixar do valor "*threshold*" de resiliência.

Deste modo, deverá considerar-se:

- Para quóruns ABD : N=2f+1+K (tendo em conta os quóruns de leitura e escrita no desenho do sistema e modelo ABD implementado)
- Para o modelo SMR-PAXOS (BFTsmart): N= 2f+1+K

  Dado f o número de réplicas falhadas a considerar. Para efeitos de implementação considerar os requisitos de avaliação experimental.

2.3.5 Outras considerações

Para obviar o processo de recuperação e transferência de estado poderá pensar-se num modelo em que existam réplicas *sentinent* sem operação ativa, prontas a serem usadas por promoção a réplicas ativas na operação do sistema. A preparação de réplicas *sentinent* pode basear-se na transferência periódica de estado correto e pode ser endereçada de forma assíncrona (background) do sistema endereçando o fator de resiliência acrescido anteriormente indicado (valor de K). Para qualquer outra questão de clarificação sobre outras opções de implementação os alunos deverão contactar os docentes.

3. Demonstração e avaliação experimental

O sistema desenvolvido deverá ser verificado na sua correção e avaliado experimentalmente. Esta avaliação deverá ser feita num ambiente distribuído no laboratório, com pelo menos 3 computadores (ou containers) interligados em rede local, com um conjunto de 3 clientes a usar o serviço DDS e 9 réplicas (sendo sempre 7 ativas e até duas sentinent), distribuídas da seguinte forma:

- Comp.1: 1 cliente, 2 réplicas + 1 réplica (sentinent)
- Comp.2: 1 cliente, 3 réplicas
- Comp.3: 1 cliente, 2 réplicas + 1 réplica (sentinent)

Teremos então um ambiente de 7 réplicas base acrescidas de duas réplicas *sentinent*. Sobre este sistema deverão ser conduzidos os seguintes testes para obtenção das seguintes métricas quantitativas sobre a correção e desempenho da implementação:

Prova de correção. Deverá ser conduzida avaliação experimental que permita verificar e argumentar que a replicação funciona corretamente e garante as propriedades desejadas (safety + liveness) seja em implementações com o sistema ABD no modelo *Read one/Write All* ou no caso da solução BFTsmart para o modelo SMR (com base na evidência das propriedades do consenso). Esta argumentação, com base na observação feira, será depois reportada em relatório.

Indicadores de *throughput* de operações suportadas na API primária (por repetição das avaliações experimentais da especificação do trabalho prático nº1 (DDS V1), agora sobre a implementação do trabalho prático nº 2 (DDS V2)). Os resultados a obter devem ser baseados na repetição dos seguintes cinco *benchmarks* (já anteriormente preparados no TP1 e que só precisam de ser repetidos para a implementação do TP2).

- *Benchmark P1*: 100 PutSet()
- *Benchmark P2*: 100 GetSet()
- *Benchmark P3*: 50 PutSet(), 50 GetSet() , com operações alternadas
- *Benchmark P4*: 50 AddElement(), 50 ReadElement , com operações alternadas
- *Benchmark P5*: *Mix* equilibrado de 100 operações, contendo as operações da API primária

Indicadores de *throughput* de operações suportadas na API estendida. Estes indicadores terão por base a nova implementação do trabalho prático nº 2 (DDS V2). Os resultados deverão ser depois representados em tabelas ou gráficos devidamente documentados com clareza e conclusões de análise crítica no relatório, o que se considerará um importante fator de avaliação comparativa dos trabalhos.

- *Benchmark E1*: 10 operações sobre cada uma das operações de pesquisa na interface estendida no caso dos dados não se encontrarem cifrados.
- *Benchmark E2*: *Mix* equilibrado de 100 operações, distribuídas por diferentes operações da API estendida com operações de pesquisa sobre dados não cifrados.
- *Benchmark E3*: 10 operações sobre cada uma das operações de pesquisa envolvendo dados cifrados para diferente processamento com base em cada algoritmo criptográfico homomórfico.
- *Benchmark E4*: *Mix* equilibrado de 100 operações, distribuídas por diferentes operações da API estendida com operações de pesquisa sobre dados não cifrados.

Os anteriores testes permitirão obter comparações para se investigar os *overheads* resultantes do suporte das operações criptográficas homomórficas, para suporte das diversas operações.

4. Indicações para entrega do trabalho

Devem ser seguidas as indicações de entrega como no caso do trabalho prático nº 1.

5. Notas sobre a avaliação do trabalho

5.1 Avaliação do trabalho do ponto de vista da implementação

O trabalho será avaliado tendo em conta as seguintes pontuações indicativas para os vários elementos, de acordo com a especificações acima referidas nos pontos 1, 2 e 3.

- Comunicação segura cliente/servidor [1 ponto]
- Comunicação segura servidor/servidor [1 ponto]
- Cobertura de todas as operações da interface (API primária) [1 pontos]
- Cobertura de todas as operações da interface (API estendida) [2 pontos]
- Integração correta do protocolo de replicação [9 pontos]
- Avaliação experimental, sua completude a análise crítica a incluir no relatório [6 pontos]

5.2 Avaliação do relatório (de acordo com as secções do template de referência)

- Organização, estilo e correção da escrita, correção conceptual, estrutura e completude, conforme *template* de referência do relatório [3 pontos]
- Qualidade da apresentação da conceção do modelo e arquitetura do sistema [4 pontos]
- Qualidade da apresentação sobre o modelo de falhas e de adversário [4 pontos]
- Completude e clareza na apresentação dos componentes da arquitetura do sistema e sua implementação [4 pontos]
- Completude, correção e clareza da apresentação de resultados da avaliação experimental bem como argumentação e análise crítica sobre os resultados obtidos face à expectativa inicial (e face à arquitetura do sistema) [5 pontos]

Avaliação final do trabalho

Será obtida pela ponderação do antes referido em 5.1 (70 %) e 5.2 (30%), com ajustamento final ponderado em 50% da análise da implementação (pacote DDS v2) bem como sua demonstração e eventual discussão e argumentação dos aspetos específicos de cada implementação.

6. Referências

Materiais das aulas teóricas de CSD 16/17 e referencias de leitura sugeridas
Ver também as referencias do trabalho prático nº 1

[1] SJHomoLib – *Synthetic Java Homomorphic Library*, TR DI-FCT-UNL e respetiva implementação disponibilizada em aula prática (laboratório)