Instituto Politécnico do Porto

Instituto Superior de Engenharia do Porto

Ignite Innovation by Upgrading Applications with Apache Storm's Latest Advancements

Blip - Blip.pt

2023/2024

1211289 Tomás Ferreira Lopes

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Ignite Innovation by Upgrading Applications with Apache Storm's Latest Advancements

Blip - Blip.pt

2023/2024

1211289 Tomás Ferreira Lopes

Licenciatura em Engenharia Informática

Abril, 2024

Orientador ISEP: Nuno Silva, nps@isep.ipp.pt

Supervisor Externo: João Reis, joao.reis@blip.pt

"I never think of the future. It comes soon enough." - Albert Einstein

Agradecimentos

Resumo

Dado o contexto em que a Blip se insere, a empresa necessita de um sistema que permita processar grandes volumes de dados em tempo real, neste caso, dados de catálogo que são recebidos de várias fontes e servem de base para a criação de eventos e informações dos mesmos. Desta forma, foi desenvolvido um sistema distribuído baseado numa arquitetura de streaming de forma a processar e armazenar os dados de forma eficiente. O objetivo deste projeto é fazer alterações no sistema existente de forma a melhorar a sua eficiência e escalabilidade, por forma a garantir que o sistema é capaz de lidar com um aumento de carga de trabalho.

Este documento apresenta o desenvolvimento de um projeto de estágio realizado na Blip no âmbito da unidade curricular de Projeto / Estágio (PESTI) da Licenciatura em Engenharia Informática (LEI) no Instituto Superior de Engenharia do Porto (ISEP).

Os resultados obtidos ...

Palavras-chave (Tema):

Processamento de dados, Sistemas distribuídos, Sistemas de tempo real

Palavras-chave (Tecnologias):

Apache Storm, Apache Kafka, Java, Nimbus, Zookeeper

Abstract

Keywords (Themes):
Data processing, Distributed systems, Real-time systems

Keywords (Technologies):
Apache Storm, Apache Kafka, Java, Nimbus, Zookeeper

Conteúdo

Lista de Figuras

Lista de Tabelas

Notação e Glossário

CD Continuous Deployment

CI Continuous Integration

CRISP-DM Cross-Industry Standard Process for Data Mining

DC Data Center

ISEP Instituto Superior de Engenharia do Porto

LEI Licenciatura em Engenharia Informática

MD Modelo de Domínio

PESTI Projeto / Estágio

UC Unidade Curricular

UKI United Kingdom and Ireland

Introdução

Este primeiro capítulo estabelece as bases necessárias para uma compreensão sólida do trabalho desenvolvido. Primeiramente é exposta a motivação do trabalho e o seu enquadramento no contexto da Blip. De seguida, são referidos os principais objetivos identificados, a abordagem adotada, os contributos da realização do projeto e uma apresentação da estrutura do documento.

1.1 Enquadramento

Este documento é o resultado do estágio desenvolvido na Blip durante o sexto semestre da Licenciatura em Engenharia Informática do ISEP no âmbito da Unidade Curricular de Projeto / Estágio (PESTI) no ano letivo de 2023/2024. A Blip é uma empresa tecnológica que pertence ao grupo Flutter que desenvolve soluções de *software* para apostas desportivas online, sendo, neste momento, o maior grupo de apostas desportivas online a nível global [4].

Desta forma, o sistema que é usado pela Blip deve estar preparado para lidar com um grande volume é bastante complexo e deve ser capaz de processar grandes volumes de dados em tempo real. Ao longo deste relatório vai ser analisado o sistema que lida com os dados de catálogo que são recebidos de várias fontes e servem de base para a criação de eventos e informações dos mesmos. Este sistema é baseado em Apache Storm e Apache Kafka, que são tecnologias que permitem processar e armazenar os dados de forma eficiente tentando evitar ao máximo que haja constrangimentos no sistema.

1.2 Descrição do Problema

O *cluster* que lida com os dados de catálogo é composto por várias máquinas que têm como função receber e transformar dados de vários eventos que são recebidos de várias fontes. Neste cluster é usado Apache Storm para fazer o processamento destes dados em tempo real.

Com o crescimento da empresa, o volume de dados que é recebido e processado tem vindo a aumentar e, além disso, nos últimos meses o grupo Flutter adquiriu uma nova empresa para a divisão de United Kingdom and Ireland (UKI) o que fez com que o volume que o *clister* tem que ser capaz de processar aumentasse substancialmente.

Desta forma, o problema que se apresenta é que este *cluster* não tem capacidade para lidar com o volume de dados para o qual está a ser sujeito e, por isso, é necessário fazer algumas otimizações por forma a não haver constrangimentos no sistema. A solução simples poderia passar por aumentar apenas a capacidade de processamento do *cluster*, mas isso seria uma solução bastante dispendiosa e, provavelmente, traria problemas de escalabilidade no futuro.

Assim, é necessário fazer uma análise ao sistema e perceber onde estão os principais problemas de performance e escalabilidade e tentar encontrar soluções que possam ser implementadas para resolver estes problemas. Numa primeira fase, é necessário perceber como o sistema está implementado na infraestrutura da empresa e fazer uma análise aos recursos usados por cada máquina que compõe o *cluster* de forma a perceber onde estão os principais problemas.

1.2.1 Objetivos

- Identificar desafios de escalabilidade e performance
- Familiarização com a ferramenta Apache Storm
- Testar e implementar Apps Storm (Java Tech Stack)
- Identificar e propor soluções de integração com Apache Kafka
- Ajudar a definir o planeamento de atualização e *rollback*

1.2.2 Abordagem

1.2.3 Contributos

1.2.4 Planeamento do Trabalho

O planeamento do trabalho concentra-se na organização e divisão do tempo útil entre as várias etapas que devem ser concluídas para de forma a atingir a solução final. A Tabela 1.1, apresenta a vista geral do planeamento elaborado.

Tabela 1.1: Planeamento do Trabalho

Etapa	**Data Início**	**Duração**
Familiarização com Apache Storm	xx/xx/2024	2 semanas
Análise das otimizações de recursos	xx/xx/2024	4 semanas
Implementação das otimizações	xx/xx/2024	6 semanas
Upgrade Apache Storm	xx/xx/2024	4 semanas

1.3 Estrutura do Relatório

O presente relatório apresenta cinco capítulos, sendo estes: Introdução, Estado da Arte, Análise e Desenho da Solução, Implementação da Solução e Conclusões.

O primeiro capítulo – Introdução – faz uma breve contextualização do projeto de forma a dar a conhecer a organização onde este foi realizado e uma descrição do problema que motivou a solução apresentada. São também explicitados os objetivos a alcançar, a abordagem a seguir, os contributos esperados, o planeamento do trabalho adotado e a estrutura do documento. Esta secção é fundamental para que o leitor consiga acompanhar o processo de desenvolvimento do projeto.

O segundo capítulo – Estado da Arte – visa realizar uma revisão de literatura, com o intuito de aprofundar assim alguns conceitos científicos e tecnologias relevantes para contextualizar o leitor no domínio teórico e prático do projeto.

O terceiro capítulo – Análise e Desenho da Solução – tem como propósito fornecer uma descrição completa do desenvolvimento da solução e como o projeto funcionará na sua totalidade, abordando tanto conceitos de domínio do problema como também os requisitos funcionais e não funcionais.

O quarto capítulo – Implementação da Solução – tem como objetivo apresentar a solução desenvolvida e descrever detalhes de implementação, assim como explicações sobre

as decisões tomadas durante o desenvolvimento do projeto, possíveis alternativas e uma avaliação geral do sistema.

O quinto, e último, capítulo – Conclusões – realiza uma síntese dos resultados alcançados com o desenvolvimento do projeto, limitações encontradas bem como uma perspetiva de futuras melhorias e considerações finais sobre o trabalho realizado.

No final do documento são também disponibilizados alguns anexos e conteúdos bibliográficos que suportam o trabalho desenvolvido apresentado ao longo do presente relatório.

Estado da Arte

O presente capítulo visa a contextualização teórica do trabalho realizado. Primeiramente, são abordados conceitos relacionados com o projeto. De seguida, é realizado um levantamento das tecnologias existentes no âmbito do projeto. Por fim, são expostas algumas soluções semelhantes já existentes no mercado, proporcionando assim uma visão ampla do contexto em que o trabalho desenvolvido se insere.

2.1 Sistemas Distribuídos

Os sistemas distribuídos são sistemas de *software* que interagem entre si através de uma rede de computadores. Estes sistemas são compostos por um conjunto de computadores autónomos que apresentam uma visão única e coerente para os utilizadores. A comunicação entre os computadores é realizada através de mensagens, permitindo a partilha de recursos e a execução de tarefas em paralelo [7].

2.1.1 Arquitetura Cliente-Servidor

Falar sobre a arquitetura cliente-servidor, conceitos e aplicabilidade.

2.1.2 Arquitetura Streaming

Falar sobre a arquitetura de streaming, Kafka, RabbitMQ, etc.

2.2 Estratégias de implantação

Introdução sobre a importância de estratégias de implantação em sistemas distribuídos. Falar sobre as estratégias mais comuns, como blue-green deployment, canary deployment,

rolling update, etc.

2.2.1 Blue-Green Deployment

Falar sobre blue-green deployment, conceitos, vantagens, desvantagens, etc. (este é o que é usado na Blip)

2.2.2 Canary Deployment

Falar sobre canary deployment, conceitos, vantagens, desvantagens, etc.

2.2.3 Rolling Update

Falar sobre rolling update, conceitos, vantagens, desvantagens, etc.

2.3 Monitorização de Sistemas

Falar sobre monitorização de sistemas, conceitos, ferramentas. Grafana, Splunk, etc.

2.3.1 Métricas

2.3.2 Registo de eventos

Análise e Desenho da Solução

Depois de contextualizados os temas e assuntos relevantes, este capítulo concentra-se na análise e elucidação do problema que sustenta o presente relatório e na apresentação do desenho da solução criada.

3.1 Domínio do Problema

3.2 Engenharia de Requisitos

A Engenharia de Requisitos é uma área muito relevanto no desenvolvimento de *software*, pois sustenta o sucesso dos projetos. Representa o processo de obtenção de requisitos através de uma análise do problema e pressupõe a definição das necessidades do cliente na procura de uma solução clara que valide a proposta. Seguindo um processo estruturado e adotando as melhores práticas, promovemos uma melhor comunicação entre as várias partes interessadas.

Considerando os aspetos mencionados, nesta secção serão apresentados todos os requisitos do sistema identificados e requisitados no início do projeto de maneira a garantir a qualidade da solução desenvolvida. Estes requisitos podem ser categorizados em funcionais - funcionalidades distintas e essenciais que o sistema deve realizar, e não funcionais - restrições impostas para que o sistema realize os requisitos funcionais corretamente.

3.2.1 Requisitos Não Funcionais

Os requisitos não funcionais não se concentram no que um sistema de software faz, mas sim em como ele funciona. São essenciais para a qualidade geral, desempenho e usabilidade do software e consideram fatores como o desempenho, a segurança, a confiabilidade e a usabilidade.

Os requisitos não funcionais apresentados em seguida, guiam-se pelo modelo FURPS+, um padrão de classificação qualitativa das características de um *software* (**F**unctionality, **U**sability, **R**eliability, **P**erformance, **S**upportability), para uma melhor experiência do utilizador. O "+"refere-se a métodos de classificação diferentes, como por exemplo, restrições de design, implementação, interface ou físicos.

Funcionalidade

- Encontram-se especificados na subsecção Requisitos Funcionais.

Usabilidade

- xxx

Confiabilidade

- A aplicação deve ser capaz de recuperar de falhas sem perda de dados.
- A aplicação deve ser capaz de recuperar de falhas minimizando o tempo de inatividade.
- A aplicação deve ser capaz de recuperar de falhas sem intervenção manual.

Desempenho

- O sistema deve ser capaz de processar transações em dia de pico sem falhas.
- O sistema deve ser capaz de processar transações em dia de pico minimizando o tempo de resposta.

Suportabilidade

- xxx

Restrições de Design

- xxx

3.2.2 Requisitos Funcionais

Os requisitos funcionais especificam as unidades e recursos funcionais de um sistema de software, concentrando-se nas funções que ele deve realizar. Eles descrevem as funcionalidades, comportamentos e operações específicas que os utilizadores devem conseguir executar podendo variar entre ações básicas, como entrada e saída de dados, a algoritmos complexos e processos de negócios.

De forma a facilitar a compreensão por parte do leitor os requisitos funcionais, encontram-se descritos, na Tabela 3.1 e na Figura 3.1, na forma de *User Stories*, seguindo a estrutura apresentada no artigo "(User) Stories for Analytics Projects – Part 1" [6].

Tabela 3.1: Requisitos Funcionais

ID	User Story
1	Como XXX, quero XXX.
2	Como XXX, quero XXX.

Figura 3.1: Diagrama de Casos de Uso

3.3 Desenho da Solução

Através da análise do problema definido, em conjunto com os requisitos funcionais e não funcionais delineados anteriormente, o principal objetivo desta secção é documentar cada fase distinta que compõe o desenho da solução idealizada.

Implementação da Solução

Este capítulo aprofunda o processo de desenvolvimento da solução para o problema do projeto, de acordo com as diretrizes estabelecidas e os princípios de design delineados no capítulo anterior. Além disso, apresenta uma análise abrangente dos resultados derivados destas decisões, enfatizando os resultados consequentes e a sua concordância com os objetivos antecipados.

4.1 Tecnologias Utilizadas

Ao longo do projeto, foram utilizadas várias tecnologias e ferramentas para a implementação da solução. Entre as mais relevantes, destacam-se as seguintes:

- **Apache Storm [2]:** Framework de processamento de *streaming* em tempo real;
- **Apache Kafka [1]:** Plataforma de mensagens distribuída;
- **Apache Zookeeper [3]:** Serviço de coordenação distribuída;
- **Nimbus [5]:** Servidor centralizado que coordena e distribui as topologias do Apache Storm;

4.2 Descrição da implementação

4.3 Testes

4.4 Avaliação da Solução

Conclusões

Este capítulo final, apresenta uma síntese dos pontos mais relevantes do trabalho desenvolvido. Em primeiro lugar, é realizada uma validação dos objetivos inicialmente propostos. De seguida, são tecidas as limitações e o trabalho futuro, visto que nenhum projeto está isento de barreiras. Na secção final, é realizada uma apreciação crítica, com o intuito de salientar pontos positivos e menos positivos no decorrer do projeto.

5.1 Objetivos concretizados

Conforme é possível verificar nos capítulos Estado da Arte, Análise e Desenho da Solução e Implementação da Solução todos os objetivos traçados na subsecção Objetivos foram atingidos na sua totalidade. A Tabela 5.1 mostra que todos os objetivos foram completamente realizados.

Tabela 5.1: Visão geral dos objetivos técnicos alcançados

Objetivo	**Grau de realização**
xxx	100%

5.2 Limitações e trabalho futuro

5.3 Apreciação final

Bibliografia

[1] Apache kafka. Disponível em `https://kafka.apache.org/`. (Acedido 07/04/2024).

[2] Apache storm. Disponível em `https://storm.apache.org/`. (Acedido 07/04/2024).

[3] Apache zookeeper. Disponível em `https://zookeeper.apache.org/`. (Acedido 07/04/2024).

[4] Blip. Disponível em `https://blip.pt/`. (Acedido 07/04/2024).

[5] Nimbus. Disponível em `https://nimbusproject.org/`. (Acedido 07/04/2024).

[6] (user) stories for analytics projects – part 1. Disponível em `https://rbranger.wordpress.com/2020/02/06/user-stories-for-analytics-projects-part-1/`. (Acedido 07/04/2024).

[7] Paulo Verissimo and Luis Rodrigues. *Distributed systems for system architects*, volume 1. Springer Science & Business Media, 2001.