UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Vitor Pelicer de Mesquita França

Arquitetura Para A Extração, Transformação E Armazenamento Em Data Warehouse Ativo

São José do Rio Preto
2024

Vitor Pelicer de Mesquita França

Arquitetura Para A Extração, Transformação E Armazenamento Em Data Warehouse Ativo

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Orientador: Prof. Dr. Carlos Roberto Valêncio

São José do Rio Preto
2024

França, Vitor Pelicer de Mesquita
F814a Arquitetura Para A Extração, Transformação E Armazenamento Em Data Warehouse Ativo / Vitor Pelicer de Mesquita França. -- São José do Rio Preto, 2024
33 p. : tabs., fotos

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (UNESP), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto
Orientador: Carlos Roberto Valêncio

1. Integração de dados. 2. Machine Learning. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Autor Vitor Pelicer de Mesquita França

Arquitetura Para A Extração, Transformação E Armazenamento Em Data Warehouse Ativo

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Comissão Examinadora

Prof. Dr. Carlos Roberto Valêncio
UNESP – Câmpus de São José do Rio Preto
Orientador

Profª. Drª. Adriana Barbosa Santos
UNESP – Câmpus de São José do Rio Preto

Profª. Drª. Renata Spolon Lobato
UNESP – Câmpus de São José do Rio Preto

São José do Rio Preto
25 de novembro de 2024

Dedico este trabalho a Deus, minha família,
professores que
fizeram parte da minha graduação e todos os colegas de que
de alguma forma ajudaram o meu desenvolvimento.

RESUMO

As arquiteturas de *Data Warehouse* convencionais fazem uso de Extração Transformação Armazenamento - ETA e com o propósito de trazer os dados de um sistema fonte para o *Data Warehouse*. Normalmente os dados são integrados por ETA em intervalos longos, geralmente de um dia e em momentos que os sistemas estão com pouca demanda de consumo. As demandas de mercado por análise de dados evoluíram significativamente e atualmente existem aplicações que requisitam por análises com alto grau de atualização, e para suportar essa demanda, passou a ser utilizadas arquiteturas de *Data Warehouse* Ativo. Nessa arquitetura o dado é constantemente atualizado, e o processo de ETA é normalmente realizado por streaming de dados ou o envio constante de micro lotes, mas devido ao alto grau de atualização, isso demanda que sejam feitas consultas e inserções ao banco de dados de maneira exaustiva, o que pode acarretar uma sobrecarga dos sistemas envolvidos. Então, este trabalho tem o objetivo de propor uma melhoria a arquitetura de integração de dados com uma estratégia para definir quando é interessante realizar as inserções dos dados no sistema, e toma a escolha baseada na relevância dos dados alterados e no seu volume. Com isto, este trabalho contribui com a redução de atualizações por meio de recurso de inteligência artificial para inferir os valores de relevância e volume necessários para acionar o processo de ETA.

Palavras–chave: *Data Warehouse* Ativo. ETA. Aprendizado por reforço. Inteligência artificial.

ABSTRACT

Conventional Data Warehouse architectures utilize Extract, Transform, and Load (ETL) with the purpose of transferring data from a source system to the Data Warehouse. Typically, data integration via ETL occurs in long intervals, often daily, during periods of low system demand. However, market demands for data analysis have evolved significantly, and there are now applications requiring analyses with a high degree of real-time updates. To meet this demand, Active Data Warehouse architectures have emerged. In these architectures, data is continuously updated, and the ETL process is usually performed through data streaming or the constant delivery of micro-batches. However, this high level of updates necessitates frequent queries and inserts into the database, which can lead to system overload. This work aims to propose an improvement to the data integration architecture by introducing a strategy to determine when it is optimal to insert data into the system. The decision is based on the relevance of the modified data and its volume. This contribution reduces the number of updates through the use of artificial intelligence to infer the relevance and volume thresholds required to trigger the ETL process.

Keywords: Active Data Warehouse. ETL. Reinforcement learning. Artificial inteligence.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ETA	Extração Transformação Armazenamento
DW	*Data Warehouse*
OLAP	*Online Analytical Processing*
OLTP	*Online Transaction Processing*
CDC	*Change Data Capture*
PPO	*Proximal Policy Optimization*

SUMÁRIO

1. Introdução

O uso de arquiteturas envolvendo DW (*Data Warehouse*) surgiram com as necessidades das corporações de atenderem a demanda crescente por análise de dados. Enquanto os sistemas na ocasião eram voltados a atender transações, sendo amplamente usados bancos de dados transacionais, e essa razão tornou crítica a necessidade do uso de uma alternativa, bancos de dados analíticos, que são mais performáticos para o processamento massivo de dados. O DW é um banco de dados analítico, em que concentramos os dados históricos para possibilitar análises mais complexas pelo usuário, é preferido por ter melhor performance para executar análises de dados do que um banco de dados transacional. O uso de DW se popularizou em conjunto com os conceitos de modelagem dimensional apresentados por Ralph Kimball, que é apresentada como uma técnica para otimizar o desempenho de consulta e análise de dados em DW (Kimball, R. and Ross, M, 2013).

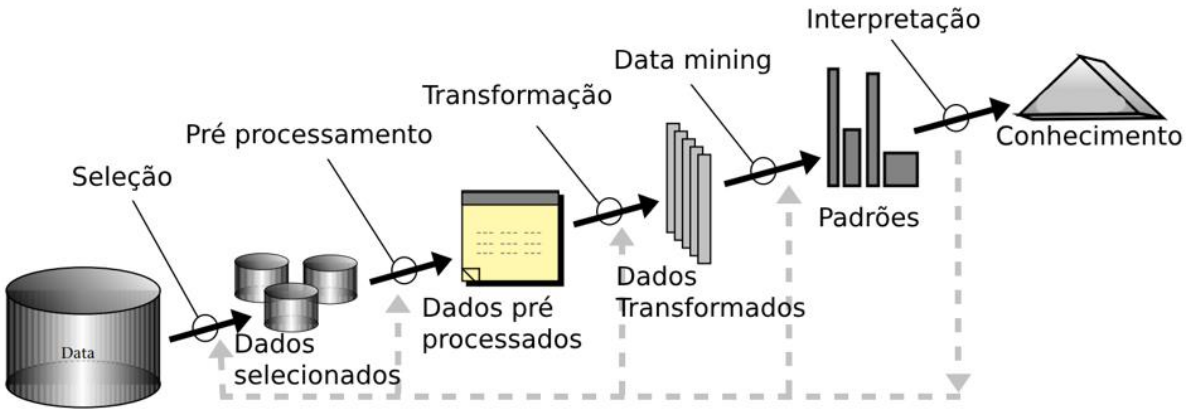
Devido as diferenças apontadas entre bancos de dados transacionais, também nomeados como OLTP (*Online Transaction Processing*) e analíticos, também nomeados como OLAP (*Online Analytical Processing*), a prática que vigorou no mercado foi a de separar o uso da aplicação e das análises. Os sistemas OLTP por serem bancos transacionais, possuem melhor desempenho ao processar transações, que incluem inserir, atualizar e deletar dados, enquanto os sistemas OLAP por serem bancos de dados analíticos possuem maior desempenho para análises complexas em grandes conjuntos de dados. Com isto, criou-se a necessidade de replicar os dados processados no OLTP ao OLAP, e a partir disso, adotou-se o uso de ETA na arquitetura para fazer a extração dos dados no OLTP, realizar as transformações necessárias, pré-processamento e modelagem dos dados e armazenar no OLAP (Kimball, R. and Ross, M, 2013).

1.1 Motivação e escopo

O processo de extração de conhecimento envolve uma série de operações, a partir dos dados até chegar no conhecimento. A seleção é a etapa em que são extraídos os dados pertinentes para a análise. O pré-processamento, é a etapa em que são feitas normalizações dos dados e melhoria da qualidade dos dados. Na transformação dos dados, os dados são adaptados para melhor se adequar a análise e algoritmo de *data mining* utilizados posteriormente. A etapa de *data mining* inclui o uso de algoritmos

para extração de padrões nos dados. E por fim, a etapa de interpretação dos padrões. O processo de extração de conhecimento está representado na figura 1. Esta análise é feita com os dados contidos no *Data Warehouse*. Os dados devem ser replicados das fontes de dados para o *Data Warehouse*, e para que a análise feita tenha valor, os dados devem estar atualizados, ou seja, devem refletir os dados presentes nas fontes.

Figura 1- Processo de extração de conhecimento



Fonte: Autor

O uso comercial dos dados passou por uma grande transformação, as grandes organizações possuem grandes necessidades relacionadas aos dados, como o fluxo, preparação, processamento e disponibilização destes dados. Assim, as grandes organizações fizeram grandes investimentos nesse setor para comportar sua crescente demanda por dados, com o objetivo de ter um sistema eficiente e volátil. A demanda por dados evoluiu conforme as novas aplicações necessitam de dados cada vez mais atualizados. Visto que a arquitetura tradicional de ETA apresenta limitações, pois apresenta desempenho restritivo diante de uma alta taxa de atualização do DW, o que não é indicado para as atuais demandas de aplicações comerciais (Sabtu et al., 2017).

Na arquitetura tradicional de ETA, a atualização dos dados é feita durante horários em que os sistemas envolvidos não são usados, por isso comumente o processo é agendado a ocorrer pela noite, e isso ocasiona na desatualização dos dados durante

o intervalo de tempo entre os processos de ETA, este problema é chamado de latência dos dados. (A. Wibowo, 2015). Devido a este problema, surgiram alternativas de arquitetura para a atualização do DW que atenda aos critérios de minimizar a latência dos dados, as principais abordagens incluem o uso de streaming de dados ou o envio de micro lotes de dados em frações curtas de tempo. E passou a se popularizar novos frameworks focados em envio e processamento de dados em tempo real, como o Apache Kafka, Apache Storm, Apache Flume, Apache Spark (S. Vyas, R. K. Tyagi, C. Jain and S. Sahu, 2021). Porém, a realização dos processos de ETA em tempo real pode trazer outros tipos de problemas, geralmente relacionados com a degradação do desempenho e a sobrecarga dos sistemas envolvidos por demandar muitas consultas e processamento de dados constante. A sobrecarga dos sistemas fonte pode fazer com que as aplicações que dependem deste serviço parar, e a sobrecarga do DW pode ocasionar na demora demasiada para executar as análises de dados (Sabtu et al., 2017).

1.2 Objetivo

Tendo em vista a importância do processo de atualização do DW para as aplicações e análises envolvidas, o objetivo deste trabalho é propor uma melhoria a arquitetura de integração de dados com uma estratégia para definir quando é interessante realizar as inserções dos dados no sistema

A contribuição deste trabalho consiste na melhora da estratégia de propagação dos dados com um algoritmo de aprendizado de máquina que permita definir os valores alvo para o acionamento do processo de ETA.

1.3 Organização da monografia

Esta monografia está organizada da seguinte maneira:

a) Capítulo 2 – Fundamentação teórica: neste capítulo são apresentados os conceitos de *Data Warehouse*, ETA, Aprendizado de máquina e os trabalhos correlatos.

b) Capítulo 3 -Metodologia e resultados: neste capítulo é apresentado a metodologia utilizada na comparação entre as arquiteturas e os seus resultados

c) Capítulo 4 – Conclusão

2. Fundamentação teórica

2.1 Considerações iniciais

Neste capítulo é apresentado a fundamentação teórica sobre os assuntos essenciais para o desenvolvimento e compreensão, além de trabalhos correlatos considerados o estado da arte. Serão apresentados conceitos vinculados a Big Data, *Data Warehouse* e arquiteturas de ETA otimizadas para atualização em quase tempo real e aprendizado de máquina.

2.2 Data Warehouse

O *Data Warehouse* foi popularizado durante a década de 1990, devido as necessidades de *Business Inteligence* (BI) e de executar rapidamente consultas analíticas com frequência. A era da *Big Data* trouxe desafios que os bancos de dados convencionais não estavam preparados (A. A. Harby and F. Zulkernine, 2022). O *Data Warehouse* é projetado para servir como um largo e complexo repositório de dados, que inclui dados de processos sensíveis a variantes no tempo, por isso o *Data Warehouse* é mantido atualizado com certa frequência definida pelo usuário, de acordo com sua demanda de uso e necessidade de atualização dos dados (N. Sakib, S. J. Jamil and S. H. Mukta, 2022). Também considerado um sistema OLAP, por ser otimizado para realizar análises sobre grandes volumes de dados, os quais um sistema OLTP não seria capaz de processar.

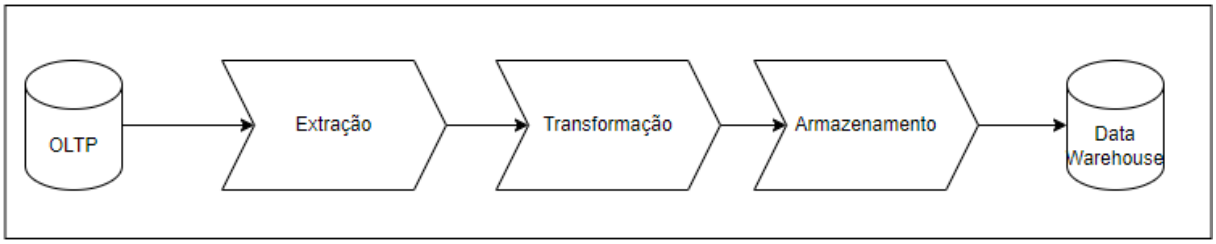
2.3 ETA

O processo de ETA, Extração Transformação Armazenamento, é uma abordagem que visa extrair dados de sistemas fontes, realizar os tratamentos necessários e inseri-los em um repositório central.

Esta abordagem é utilizada para realizar a atualização dos dados no sistema DW, que realiza a inserção dos novos dados. Esse processo envolve um ou mais sistemas fonte, onde são extraídos os dados, comumente são bancos de dados (OLTP). Após a extração, os dados são remodelados, normalmente seguindo o modelo dimensional e inseridos em um repositório único (DW) (Kimball, R. and Ross, M, 2013). O processo de ETA é agendado para ser executado com um intervalo de tempo, comumente em horários em que o sistema não é usado, para não gerar uma

sobrecarga junto ao uso dos analistas e cientistas de dados. O processo é comumente executado a noite, e entre as execuções do ETA, o Data Warehouse fica desatualizado em relação a fonte. Na figura 2 é representado o processo de ETA descrito.

Figura 2- Processo de ETA



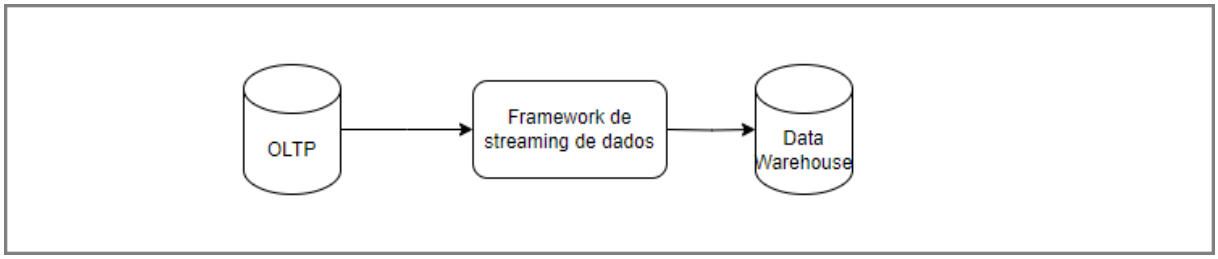
Fonte: Autor

2.4 *Data Warehouse* Ativo

A arquitetura de um DW Ativo tem sua estrutura semelhante a arquitetura de um DW convencional, com a diferença da necessidade de atualização frequente, então utiliza-se de técnicas para ir inserindo os dados no DW em quase tempo real, logo o intervalo entre as atualizações muda de dias para horas ou minutos. Como exemplos, as técnicas mais abordadas envolvem o uso de *frameworks* de *streaming* de dados, técnicas de *Change Data Capture* (CDC). No caso o CDC, existem abordagens baseadas em carimbo de data/hora, baseada em log e baseada em eventos. Na abordagem baseada em log, é feito o monitoramento de logs do sistema fonte para detectar quando há uma inserção ou alteração dos dados para então acionar o processo de ETA.

O uso de *streaming* de dados para realizar o ETA é comum em aplicações em que o tempo de latência do dado deve ser minimizado ao máximo, sendo muito aplicado a redes sociais e sistemas de *IoT* por exemplo (S. Vyas, R. K. Tyagi, C. Jain and S. Sahu, 2021). Neste sistema, quando o dado é inserido, o framework faz o post dos novos dados em tópicos e o *Data Warehouse* atua como consumidor do *streaming*.

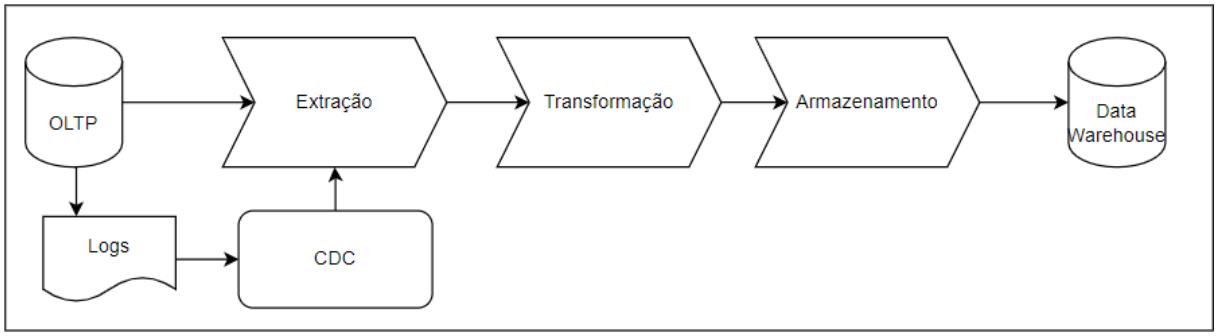
Figura 3: Arquitetura de *Data Warehouse* Ativo usando *streaming* de dados



Fonte: Autor

O CDC é utilizado para armazenar as alterações realizadas no sistema fonte, realizado a partir do monitoramento dos logs do banco de dados, o que possibilita executar o ETA somente nas informações necessárias. Uma das vantagens desta técnica é gerar um conjunto de dados ideal para o carregamento, o que pode ajudar e evitar sobrecargas ao DW, mas pelo outro lado, as técnicas de CDC exigem componentes extra para o seu funcionamento, o que aumenta a complexidade da sua arquitetura, componentes tais como uma tabela de log, ou parâmetros para acompanhar o processo ao utilizar identificadores de novos registros. (Gorhe, Swapnil, 2020). Na figura 4 é representado a arquitetura de ETA baseado em CDC. O processo de CDC armazena as alterações a partir dos *logs* do banco de dados de origem dos dados, o que possibilita o processo de ETA inserir apenas os dados alterados para o DW.

Figura 4: Arquitetura de *Data Warehouse* usando CDC.



Fonte: Autor

2.5 Aprendizado de Máquina

O Aprendizado de máquina, é um subcampo da inteligência artificial voltado ao desenvolvimento de modelos capazes de realizar previsões e decisões baseadas nos dados. É dividido entre três tipos principais, aprendizado supervisionado, não

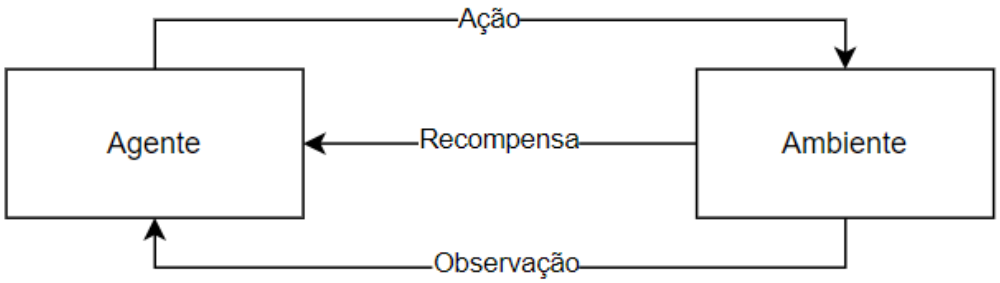
supervisionado, e por reforço (T. R. N and R. Gupta, 2020). Aqui será dado maior aprofundamento para o aprendizado por reforço, por ser o enfoque do trabalho.

Para os modelos supervisionados, é necessário o treinamento, pois o objetivo do aprendizado supervisionado é a partir dos dados de treinamento produzir uma função completa que poderá ser utilizada em novas instâncias. É um algoritmo capaz de analisar e generalizar corretamente. Sendo assim, nesse modelo, dado uma entrada e uma variável de saída, o algoritmo irá ser capaz de generalizar corretamente para outros casos.

No modelo não supervisionado, os dados não são rotulados, apenas existe a variável de entrada, sem a variável de saída. Dessa forma o aprendizado de máquina se encarrega de inferir um modelo com base nos padrões dos dados ou inferir um modelo que tenha a mesma densidade e probabilidade dos dados de entrada.

Aprendizado por reforço é um paradigma para treinar agentes inteligentes e tomar decisões complexas em ambientes dinâmicos. Consiste na capacidade de interagir com o ambiente e aprender com base nisso para otimizar as suas ações se baseando nas recompensas retornadas. De maneira diferente do aprendizado não supervisionado, os agentes não precisam de dados catalogados com respostas. Os agentes de aprendizado por reforço aprendem a partir da tentativa e erro, ao tomar ações desconhecidas durante o treinamento, recebem recompensas ou penalidades baseadas em suas ações. A figura 5 representa as interações entre o agente e o ambiente. (A. del Rio, D. Jimenez and J. Serrano, 2024)

Figura 5: Interações do aprendizado por reforço.



Fonte: Autor

O ambiente representa o sistema ao qual o agente interage, este pode representar diversas aplicações, podendo ser sistemas complexos, ou interações do

mundo real, como através de visão computacional ou processamento de linguagem natural, existe muita flexibilidade para se definir um ambiente. Mas deve ser bem definido o passo, que é o processo do ambiente em relação a uma ação, o que atualiza seus estados, retorna uma recompensa, e a observação dos estados, e uma função de recompensa para alimentar o agente. O agente é o modelo de aprendizado de máquina em funcionamento, em que recebe as recompensas e as observações (que são os estados do ambiente que o agente tem acesso) e retorna uma ação.

Existem vários modelos de aprendizado de máquina, dentre os mais populares, existe o *Q-learning*, em que objetivo é aprender uma função de valor de ação ótima, ou seja, Q*. DQN (*Deep Q-Network*), que combina o aprendizado por reforço do *Q-learning* com uma rede neural profunda (T. R. N and R. Gupta, 2020). O PPO (*Proximal Policy Optimization*), que tem como objetivo atualizar a sua política através de mudanças no gradiente durante o treinamento A. del Rio, D. Jimenez and J. Serrano, 2024).

Um conceito importante de considerar para o treinamento de um modelo são os hiperparâmetros, que são variáveis de entrada no modelo que afetam como o modelo aprende. Esse conceito interfere muito no desempenho do modelo.

2.6 Trabalhos correlatos

Existem diversos trabalhos correlatos na literatura relacionados a arquiteturas de DW ativo, sendo que estes trabalhos abordam diferentes problemas na arquitetura. Dessa forma, não existe um consenso em relação a uma arquitetura de DW ativo, já que cada proposta privilegia uma solução de um problema em detrimento de outro. Os desafios mais abordados na literatura para a atualização de dados em DW em quase tempo real com arquitetura baseada em streaming de dados são listados por ERUM MEHMOOD e TAYYABA ANEES (E. Mehmood and T. Anees, 2020):

- Manter o OLAP disponível e performático.
- Unir o uso de processamento em fluxo distribuído e um sistema DBMS.
- Carregamento eficiente do fluxo de dados.
- Lidar com fluxos de dados repetidos.
- Manter um baixo uso de memória, mesmo com altos fluxos de dados.
- Estratégia de carregamento de blocos de dados relacionais baseado em disco.

- Integração de fontes de dados heterogêneas, sobrecarga de fonte de dados, sobrecarga de dados mestre, bancos de dados sem esquema
- Disponibilidade contínua de bancos de dados em caso de DWH distribuído

Para Adilah Sabtu (Sabtu et al, 2017), o sistema com ETA quase em tempo real tem como as principais características a serem mantidas:

- Alta disponibilidade – O fato da atualização constante dos dados implica que qualquer falha aos sistemas envolvidos pode afetar as operações. Portanto, se houver alguma indisponibilidade no sistema, os dados já deixarão de estar atualizados no DW.
- Baixa Latência – As necessidades de negócio para dados frescos exigem que os dados sejam atualizados rapidamente e constantemente. O tempo de demora na atualização dos dados é chamado de latência. As arquiteturas tradicionais de ETL não acomodam essa necessidade, já que nesse modelo as atualizações são programadas entre grandes espaços de tempo.
- Escalabilidade horizontal – A escalabilidade do sistema é o que poderá melhorar a performance do sistema e tornar este mais resiliente a falhas, diminuindo o risco de interrupção

Para estas características a serem mantidas, Adilah Sabtu listou os desafios relacionados a cada característica em cada etapa do ETA. Durante a extração, possíveis problemas são a heterogeneidade dos dados fonte, necessidade de backup, inconsistências do OLAP e sobrecarga do sistema fonte. Durante a transformação, foram apontados problemas com overhead dos dados principais, uso de servidores para agregar os dados. Para a fase de armazenamento foram apontados problemas com inconsistências internas no OLAP e degradação da performance.

No estudo de Kamal (Kakish, Kamal & Kraft, Theresa, 2012) são apontados algumas possíveis técnicas para utilizar na arquitetura de DW ativo:

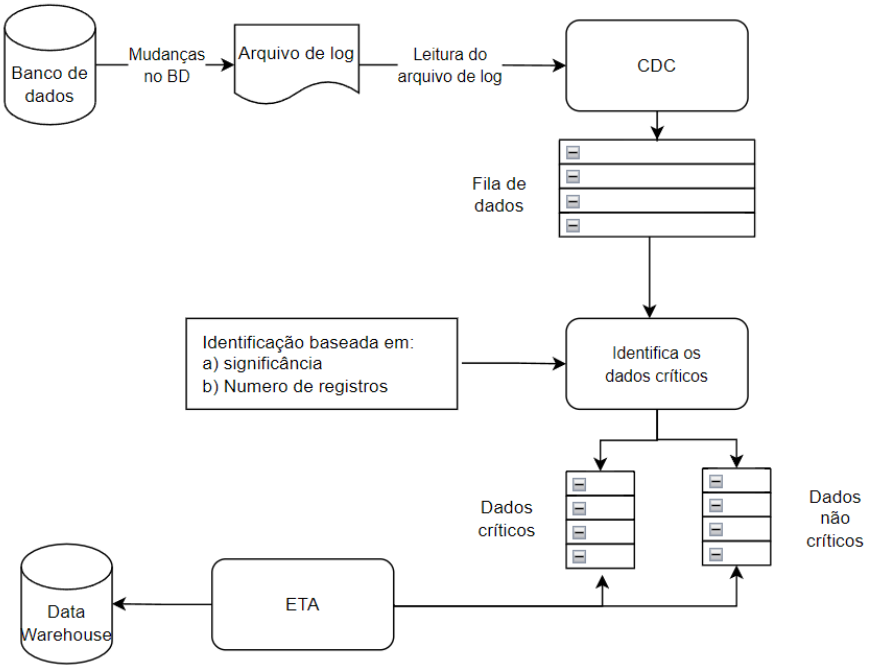
- Módulos reguladores de fluxo – Executa o papel de controlar o que é extraído do sistema fonte e o que é inserido no DW. Permitindo avaliar se as mudanças a serem propagadas são relevantes e se as fontes de dados estão disponíveis para transmitir os dados. A vantagem dessa técnica é ter controle do fluxo de dados, então adotada para evitar a sobrecarga dos sistemas envolvidos por consultas e inserções.

- Trickle and Flip e External Real-Time Data Cache – É uma técnica que envolve usar tabelas de estágio para inserir os dados em tempo real, o que as torna disponíveis a consulta, e posteriormente inseridos no DW. A vantagem dessa técnica é que evita a degradação da performance do DW ao mesmo tempo que mantem os dados disponíveis para análise.

A arquitetura proposta por Vilela e Rodrigues (F. de Assis Vilela and R. Rodrigues Ciferri, 2021), o *Data Magnet* é uma arquitetura feita para inserir dados em um DW ativo de forma a evitar alguns dos problemas aqui já citados. Nessa arquitetura é utilizado streaming de dados, onde na criação dos dados, os dados são enviados para o broker (servidor) com uma tag que indica o tipo de dado que faz parte. Dessa forma, a extração dos dados é isolada do sistema OLTP, por não acessar os dados nesse sistema esta arquitetura é dita não intrusiva. E os dados são inseridos através de uma integração do consumidor do streaming e o DW. Este modelo também é vantajoso por ser reativo, ou seja, no momento da criação das alterações nos dados, o sistema é acionado, e envia os dados para o broker. Porém ainda existem problemas com esta proposta de arquitetura, já que não elimina por completo os problemas envolvendo o streaming de dados já citados. Essa abordagem pode gerar sobrecarga no sistema DW devido a grande quantidade de inserções, dificuldades de integração dos sistemas com o framework de streaming e dificuldades de carregamento eficiente do fluxo de dados, visto que transformações complexas podem exigir muito processamento do sistema de streaming.

Na arquitetura proposta por Tanvi (Tanvi Jain, Rajasree S, Shivani Saluja, 2012), é apresentada uma forma de controlar o fluxo de dados que se baseia na relevância e número de registros a serem inseridos, de modo que os dados mais relevantes são inseridos no DW com prioridade. O uso de CDC na arquitetura cria uma fila de dados a serem inseridos, e posteriormente estes dados são classificados como dados críticos e não críticos em duas filas distintas. Dessa forma o processo de ETA faz leitura das duas filas de dados e atualiza apenas os dados críticos em tempo real, enquanto os dados não críticos esperam por mais tempo até serem inseridos no DW. Na figura 6 está uma representação desta arquitetura.

Imagem 6: Arquitetura proposta por Tanvi.

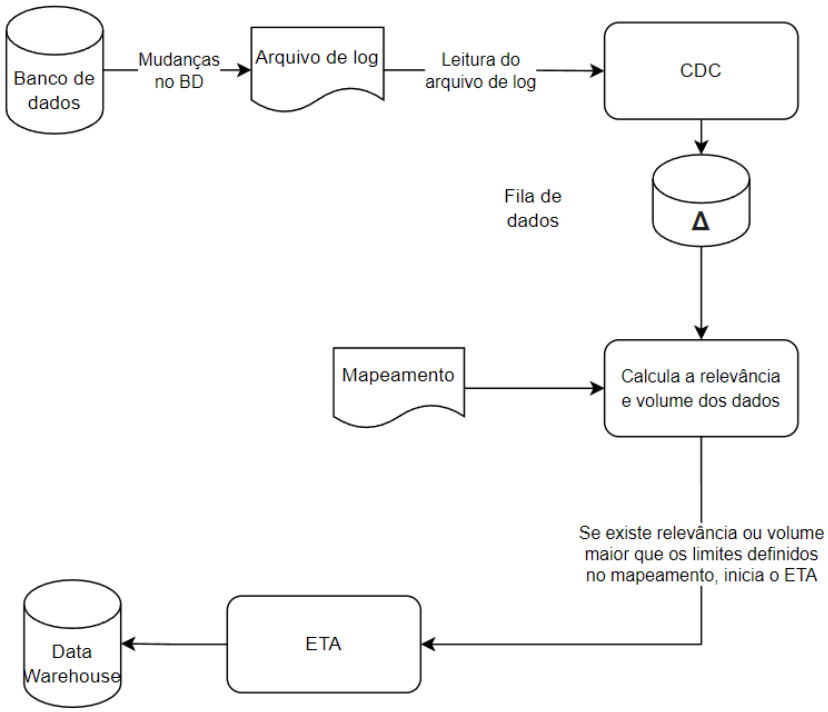


Fonte: Autor

Legenda: Este diagrama representa a arquitetura de um *Data Warehouse* ativo com diferença de dados críticos e dados não críticos.

Na arquitetura proposta por Paulo (SCARPELINI, Paulo, 2013), ETA PoCom, também é proposto uma arquitetura que controla o fluxo de dados, que se baseia na relevância e volume dos dados a serem inseridos. As alterações feitas no banco de dados de origem são armazenadas nos logs, que por meio de CDC guarda essas informações em uma tabela chamada delta. A cada período pré-estipulado é calculado a relevância e o volume de dados, e se os dados a serem movidos atingem a relevância ou volume mínimo, todos os dados são movidos, dessa forma é reduzido o número de inserções, e o DW se mantem com dados relevantes para análise. A figura 7 representa a arquitetura ETA-PoCom:

Figura 7: Arquitetura ETA-PoCom.



Fonte: Autor

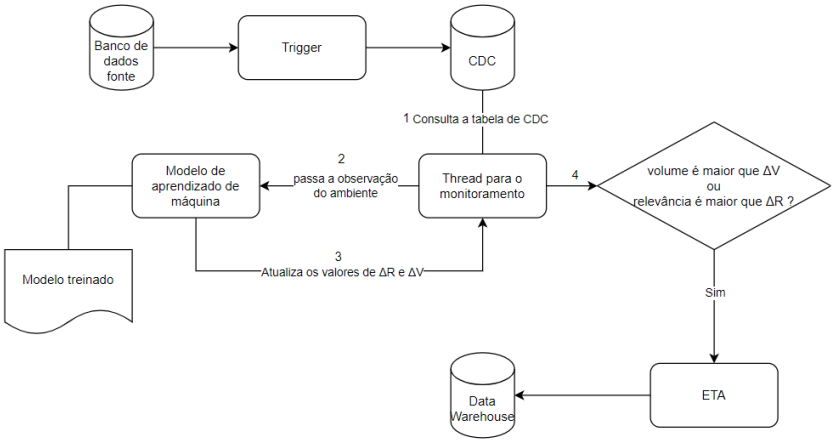
2.7 Arquitetura proposta

A arquitetura proposta neste trabalho é semelhante as descritas por Paulo (SCARPELINI, Paulo, 2013) e Tanvi (Tanvi Jain, Rajasree S, Shivani Saluja, 2012) no uso de métricas de volume e relevância para regular o fluxo de dados entre o sistema fonte e o DW. O processo utiliza de aprendizado de máquina para determinar quais são os valores mínimos para ativação do processo de integração. Essa inteligência artificial utiliza o aprendizado por reforço. O algoritmo de aprendizado por reforço aprende como comportar-se para obter a soma máxima de recompensas, o treinamento é feito após várias observações na fase de exploração do algoritmo, etapa em que o algoritmo usa as observações do ambiente, que nessa arquitetura são os dados do sistema fonte e retorna uma ação aleatória que é posta a prova no ambiente e posteriormente retorna uma recompensa que pode ser positiva ou uma penalidade. Após a etapa de exploração, o modelo salva os dados da política gerada pelo treinamento e está pronto para atuar escolhendo os valores de ativação da integração para relevância e volume, tais valores também identificados como ΔR e ΔV respectivamente.

A arquitetura é baseada no uso de estratégia de *change data capture* (CDC), que é uma estratégia de migrar apenas as alterações feitas no sistema fonte ao DW. Existem diferentes formas de consultar somente as alterações do banco de dados, entre as opções, existem CDC utilizando os logs do banco de dados, também chamados pela sigla WAL (*Write Ahead Logs*), adição de coluna com data e hora da última modificação para cada tabela monitorada, e uso de triggers no banco de dados. O sistema pode ser adaptado para usar qualquer forma de CDC, mas o método adotado aqui é o uso de *triggers*, que são gatilhos acionados quando alguma transação é feita no banco de dados para uma tabela específica, e então cria um registro da alteração em uma tabela própria para o CDC.

Na arquitetura é utilizado um *thread* para monitorar cada tabela mapeada, este processo monitora a cada período as alterações feitas na tabela de CDC, realiza os cálculos de relevância e volume para as alterações e compara com ΔR e ΔV. Caso as alterações presentes na tabela de CDC tenham relevância e volume superiores a ΔR e ΔV, então o thread atualiza o DW com as alterações e apaga o registro de alterações da tabela de CDC. A figura 8 representa a arquitetura proposta:

Figura 8: Arquitetura ETA proposta.



Fonte: Autor

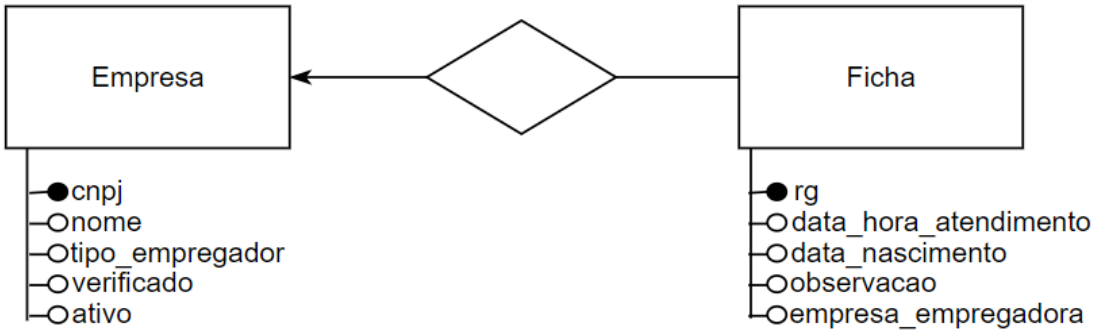
O diferencial da arquitetura proposta é a autorregulagem dos valores de ΔR e ΔV por parte do modelo de aprendizado de máquina, e a flexibilidade que permite implementar uma função de recompensa diferente ao modelo. O modelo consegue aprender com base nos dados reais, uma forma de otimizar as suas recompensas por meio de ações ao variar os valores de ΔR e ΔV.

3. Metodologia e resultados

Neste trabalho são apresentados os conceitos de DW e os seus problemas que envolvem a atualização devido a necessidade das novas aplicações em dados constantemente atualizados. Para isso, o enfoque deste trabalho é apresentar uma arquitetura para DW Ativo de forma a resolver ou mitigar os problemas de desempenho ligados a inserções frequentes no DW. Este trabalho é um aprimoramento das contribuições de Paulo (SCARPELINI, Paulo, 2013) e Tanvi (Tanvi Jain, Rajasree S, Shivani Saluja, 2012), em que o diferencial é o uso de um modelo de aprendizado de máquina para definir os valores de acionamento do sistema, o que resolveria um dos problemas citados por Paulo, que é a dificuldade da escolha dessas variáveis que impactam tanto o desempenho do sistema. Como comparação entre os dois sistemas, são implementadas e testadas as duas arquiteturas, a proposta neste trabalho e a proposta por Paulo, e feita a comparação da evolução da relevância e volume ao longo do tempo em ambos os sistemas. Nos resultados espera-se que a arquitetura proposta gere alguma economia de desempenho ao DW ao realizar menos inserções, mantenha o DW atualizado com dados relevantes e consiga manter os valores dos deltas em pontos que mantenham o equilíbrio de ativações por volume e relevância automaticamente.

O modelo de dados utilizado representa o cadastro de empresas e empregados, foi utilizado uma tabela para empresas e uma tabela para ficha de trabalhadores. O modelo de dados corresponde ao da figura 9, é composto por uma tabela de ficha para os empregados e uma tabela para as empresas. Existe um relacionamento entre empresas e as fichas, em que as fichas indicam a qual empresa o empregado trabalha.

Figura 9: Modelo de dados utilizado



Fonte: Autor

Legenda: O diagrama representa o modelo de dados que inclui as tabelas Empresa e Ficha, que possui uma relação 1:n.

Os testes foram realizados em um computador com um processador intel core i5 de 12ª geração e 16GB de memória RAM, utilizando o sistema operacional Windows 11. Para a inserção contínua dos dados no banco de dados fonte, foi utilizado um script gerador com capacidade de gerar dados aleatórios, mas que atendam as características de cada campo da tabela, o que os tornam extremamente verossímeis. As taxas de inserção e atualização dos dados por segundo foram definidas no script. Sendo criado ou atualizado 7 empresas por segundo e 12 fichas por segundo. Também contém um limite de 200 inserções de empresas e um limite de 1000 inserções de fichas no sistema. Após inseridas as 200 empresas e as 1000 fichas, o script apenas faz atualizações no banco de dados. E como os dados são gerados de forma pseudoaleatória, garante-se que os mesmos dados foram usados entre os testes, já que foi configurado a mesma seed de criação para os dados. Para o treinamento do modelo, foram inseridas as taxas de crescimento dos dados no ambiente utilizado pelo aprendizado de máquina, para que fosse possível realizar o treinamento de maneira rápida, com um número grande de passos no ambiente, o que aumenta as chances do modelo aprender de maneira eficiente. O modelo utilizado foi o PPO, com os hiperparâmetros definidos com Gamma = 0,5 e 2048 passos por atualização do modelo. Foram realizados 10 000 passos no ambiente para o treinamento dos modelos dos testes realizados.

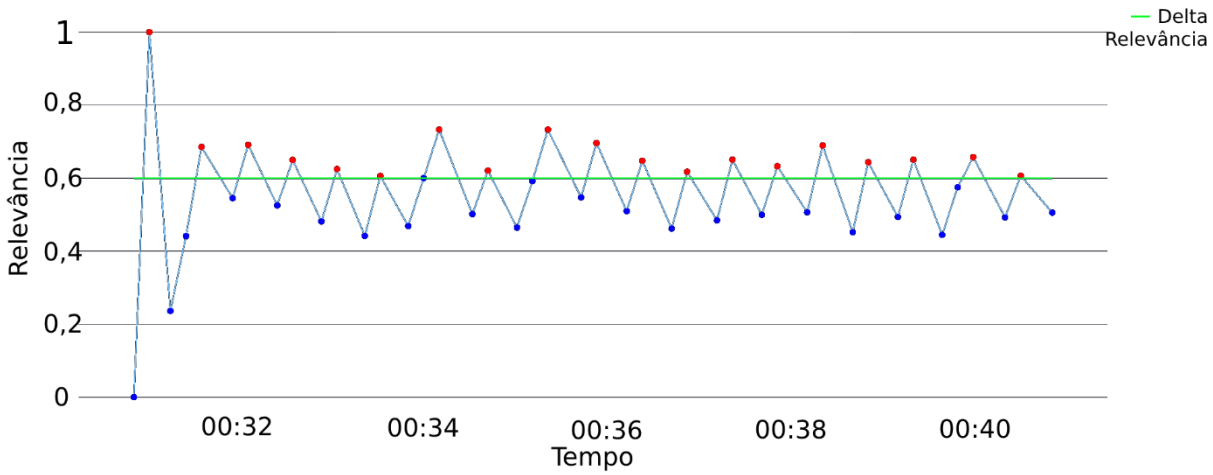
A discussão dos resultados é dividida entre três partes, análise da arquitetura apresentada por Paulo (SCARPELINI, Paulo, 2013), ETA Po-com, onde se mostra

quais os pontos críticos da aplicação, a arquitetura proposta neste trabalho com uma função de cálculo de recompensa do modelo de aprendizado de máquina que visa igualar o número de ativações por volume e relevância e outro teste utilizando um cálculo de relevância diferente, que visa maximizar o produto entre volume e relevância ao mesmo tempo que visa igualar o número de ativações por volume e relevância.

3.1 Teste da arquitetura ETA Po-com

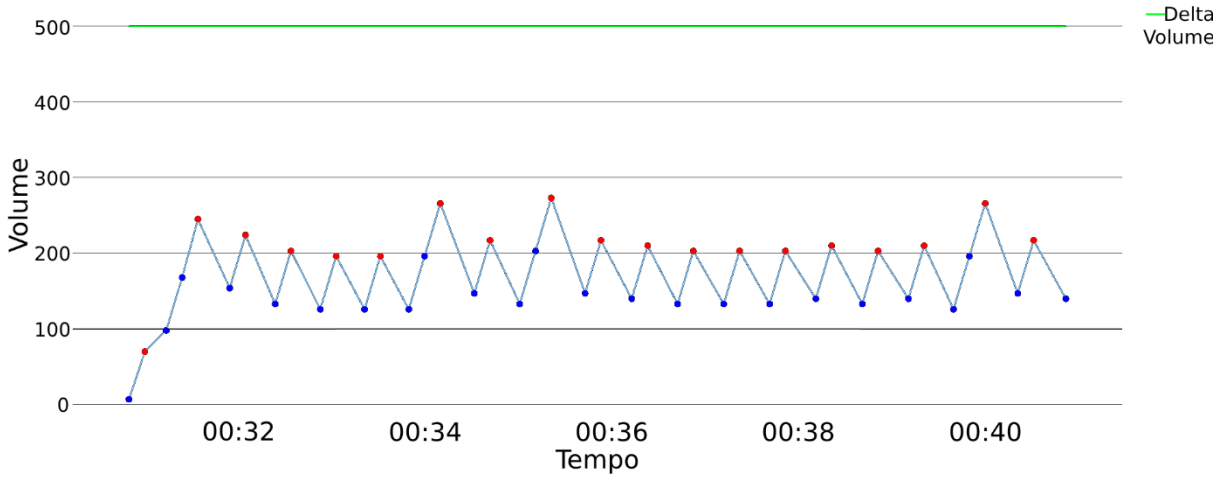
A arquitetura ETA Po-com foi implementada e posta à prova, no ambiente mencionado. O monitoramento dos dados e ativações foi feita usando dois gráficos. No teste feito, utilizou-se os valores de ΔV=500 e ΔR=0.6, e os resultados estão contidos nas figuras 10 e 11, na qual a linha azul representa a evolução da relevância ou volume, e a linha verde representa o valor do delta. Os pontos presentes nas linhas representam uma observação do algoritmo, em que um ponto azul representa uma observação que não resultou em ativação, e o ponto vermelho representa uma observação que resultou em ativação.

Figura 10: Gráfico relevância x tempo para o ETA Po-com.



Fonte: Autor

Figura 11: Gráfico volume x tempo para o ETA Po-com.



Fonte: Autor

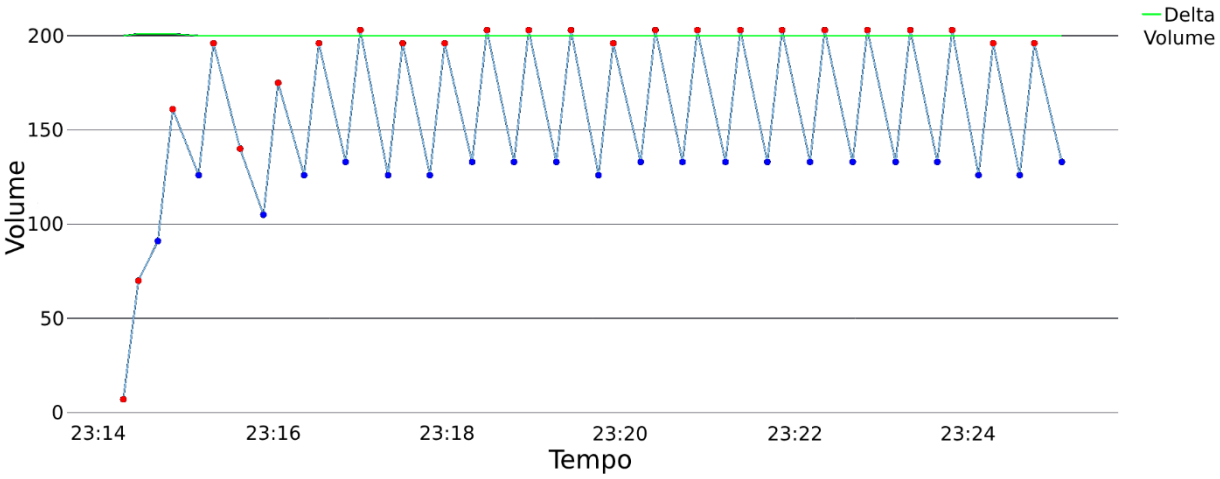
Legenda: No gráfico a linha com ponto representa a evolução do volume dos dados, os pontos em vermelho representam uma ativação do ETA, e a linha sem pontos representa o valor de ΔV.

No resultado do experimento, a causa dos acionamentos do ETA foram todas ligadas a relevância, o que mostra o peso da configuração dos valores de ΔR e ΔV tem no desempenho do sistema. Demonstra-se que o cálculo de volume foi subutilizado nesse caso, já que não acionou o processo de ETA nenhuma vez. Para melhorar o desempenho desta arquitetura, podem ser testadas outras combinações de ΔR e ΔV manualmente, e caso as taxas de crescimento de volume ou relevância mudem, o sistema teria de ser reconfigurado.

3.2 Teste da arquitetura proposta caso 1

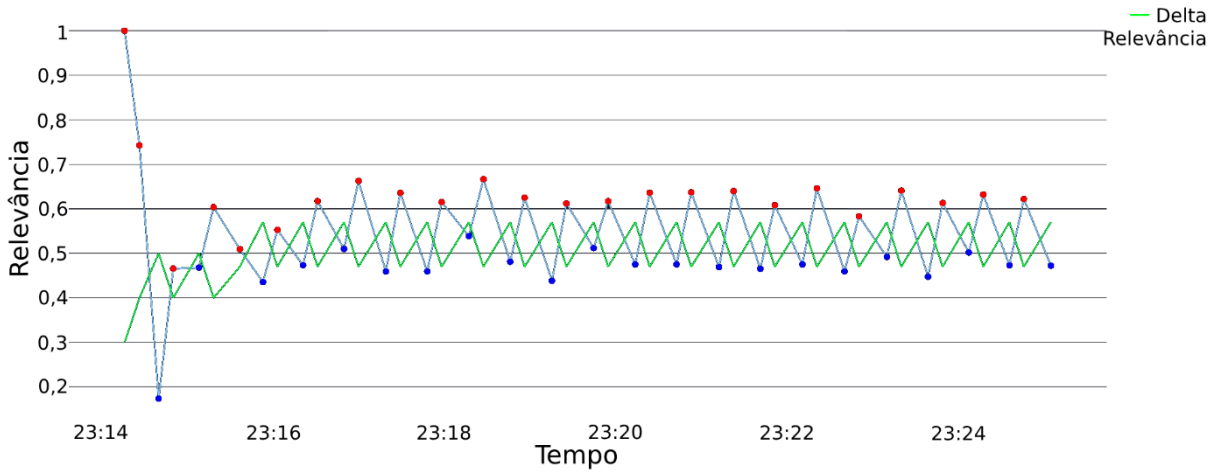
Neste teste da arquitetura proposta, foi utilizado uma função de cálculo da recompensa para aproximar o número de ativações por volume e o número de ativações por relevância. O que torna uma métrica interessante, já que evita a subutilização de uma das métricas. Os resultados estão contidos nas figuras 12 e 13.

Figura 12: Gráfico volume x tempo para o ETA proposto, caso 1.



Fonte: Autor

Figura 13: Gráfico relevância x tempo para o ETA proposto, caso 1.



Fonte: Autor

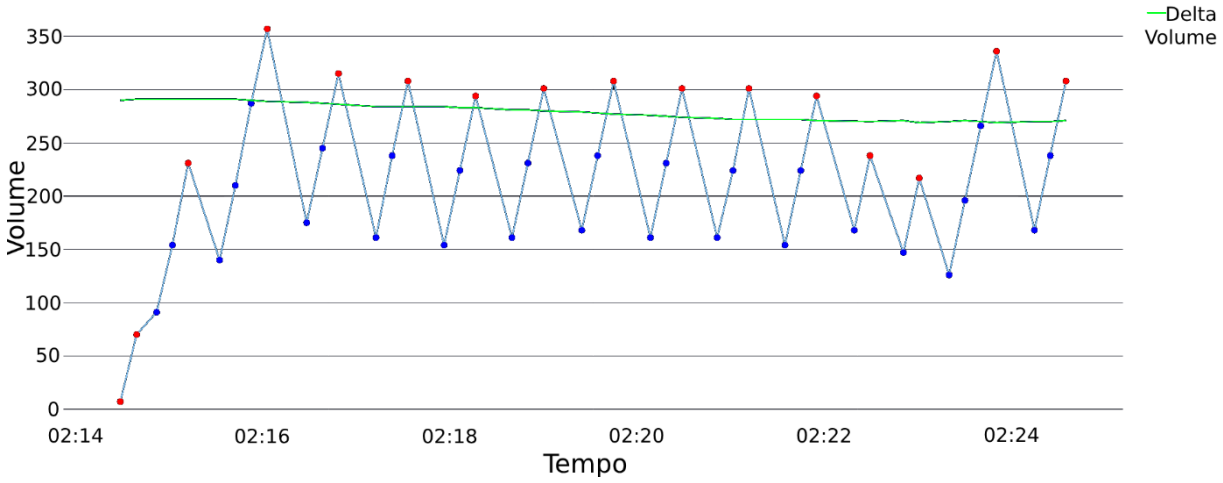
A partir dos gráficos, percebe-se que o algoritmo conseguiu buscar uma estabilidade e fez com que a maioria das ativações sejam ativações mútuas, sendo ativadas tanto por volume quanto relevância, o que atende a ideia da função de recompensa utilizada em buscar o equilíbrio no número de ativações entre volume e relevância.

3.3 Teste da arquitetura proposta caso 2

Neste teste da arquitetura proposta, foi utilizada uma função de cálculo da recompensa que visa maximizar o produto de volume e relevância nos acionamentos

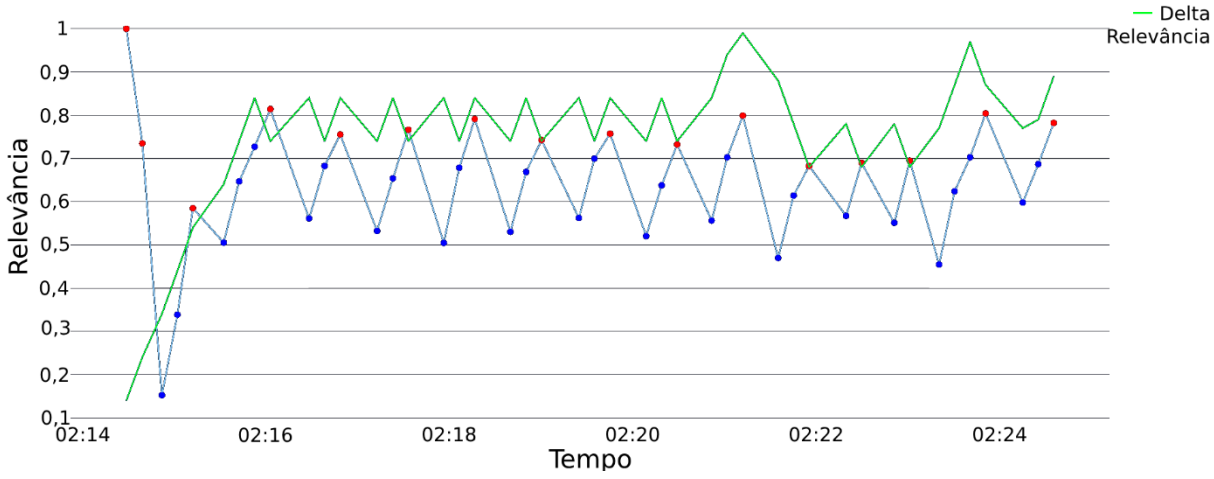
e ao mesmo tempo busca igualar a média de ativações. Assim como no teste anterior, penaliza diferenças grandes do delta para o valor real. Essa métrica de cálculo da recompensa é vantajosa por aproximar os pontos de ativação do ETA em pontos que a relevância e o volume são altos. Os resultados estão contidos nas figuras 14 e 15.

Figura 14: Gráfico volume x tempo para o ETA proposto, caso 2.



Fonte: Autor

Figura 14: Gráfico relevância x tempo para o ETA proposto, caso 2.



Fonte: Autor

A partir dos gráficos, percebe-se que o algoritmo conseguiu seguir os objetivos propostos pela função de cálculo da recompensa. Nos pontos em que houve a ativação do ETA, os deltas estão bem próximos dos valores reais, e os números de

ativações por volume e relevância não são muito discrepantes, havendo 11 ativações por volume e 9 ativações por relevância.

4. Conclusão

O objetivo do trabalho foi propor uma melhoria a arquitetura de integração de dados com uma estratégia para definir quando é interessante realizar as inserções dos dados no sistema. Com os valores da política de atualização definidos pelo modelo de aprendizado de máquina, torna mais fácil para o usuário e o modelo consegue se adaptar a novos padrões do gráfico via retreinamento.

A fim de fundamentar o trabalho, foram apresentados trabalhos correlatos, e mostradas características importantes, principalmente dos desafios enfrentados com as atualizações em tempo real e da possível solução com o uso da priorização de dados sensíveis ao sistema. Esse método permite reduzir a taxa de atualizações, mas ainda mantem o sistema atualizado em tempo real, com uma política de atualização baseada em volume e relevância.

A estratégia de usar um modelo de aprendizado de máquina para inferir os melhores valores para a política de atualização é uma solução viável para um problema que os trabalhos correlatos não solucionam. Caso haja uma mudança nas taxas de modificações dos dados no sistema fonte, o modelo é capaz de acompanhar as mudanças, e até mesmo ser retreinado para e readequar a um novo padrão. O retreinamento com base nos novos dados pode ser feito de maneira paralela ao funcionamento do sistema, mas uma dificuldade a se enfrentar é o número alto de passos que o modelo tem que dar para ser treinado, o que pode ser muito demorado para realizar o treinamento. Existem duas soluções, o armazenamento do histórico da tabela de CDC, na qual o modelo pode iterar pelo histórico de modificações e realizar o treinamento do modelo ou realizar o treinamento do modelo com base apenas nas taxas atuais de crescimento de volume e relevância. A tabela 1 contém uma comparação entre as arquiteturas de integração de dados citadas neste trabalho.

Tabela 1: Comparação entre arquiteturas.

	Tanvi	Po-Com	Arquitetura proposta
Quase tempo real	Sim	Sim	Sim
Priorização de relevância e volume	Sim	Sim	Sim
Adaptativo	Não	Não	Sim

Fonte: Autor

Após analisar os resultados, a conclusão é que a arquitetura conseguiu diminuir o número de inserções no DW em 16% entre o ETA-PoCom e o teste do segundo caso da arquitetura proposta, e permitiu ser adaptável para diferentes objetivos com a política gerada pelo modelo de aprendizado de máquina.

5. Bibliografia

A. A. Harby; F. Zulkernine. **From data warehouse to lakehouse: a comparative review.** In: IEEE International Conference on Big Data (Big Data), 2022, Osaka. Proceedings... Osaka: IEEE, 2022. p. 389-395. DOI: 10.1109/BigData55660.2022.10020719.

DEL RIO, A.; JIMENEZ, D.; SERRANO, J. **Comparative analysis of A3C and PPO algorithms in reinforcement learning: a survey on general environments.** In: IEEE Access. [S. l.: s. n.], 2024. DOI: 10.1109/ACCESS.2024.3472473.

DE ASSIS VILELA, F.; RODRIGUES CIFERRI, R. **A novel solution to perform real-time ETL process based on non-intrusive and reactive concepts.** In: International Conference on Computational Science and Computational Intelligence (CSCI), 2021, Las Vegas. Proceedings... Las Vegas: IEEE, 2021. p. 556-561. DOI: 10.1109/CSCI54926.2021.00158.

GORHE, S. **ETL in near-real-time environment: a review of challenges and possible solutions.** [S. l.], 2020.

JAIN, T.; S, R.; SALUJA, S. **Refreshing datawarehouse in near real-time.** International Journal of Computer Applications, [S. l.], v. 46, n. 18, p. 24-29, maio 2012. DOI: 10.5120/7042-9482.

KAKISH, K.; KRAFT, T. **ETL evolution for real-time data warehousing.** [S. l.], 2012.

KIMBALL, R.; ROSS, M. **The data warehouse toolkit: the complete guide to dimensional modeling.** 3rd. ed. Indianapolis: John Wiley & Sons, Inc., 2013.

MEHMOOD, E.; ANEES, T. **Challenges and solutions for processing real-time big data stream: a systematic literature review.** In: IEEE Access, [S. l.], v. 8, p. 119123-119143, 2020. DOI: 10.1109/ACCESS.2020.3005268.

SAKIB, N.; JAMIL, S. J.; MUKTA, S. H. **A novel approach on machine learning based data warehousing for intelligent healthcare services.** In: IEEE Region 10 Symposium (TENSYMP), 2022, Mumbai. Proceedings... Mumbai: IEEE, 2022. p. 1-5. DOI: 10.1109/TENSYMP54529.2022.9864564.

SABTU et al. **The challenges of extract, transform and loading (ETL) system implementation for near real-time environment.** In: International Conference on Research and Innovation in Information Systems (ICRIIS), 2017, Langkawi. Proceedings... Langkawi: IEEE, 2017. p. 1-5. DOI: 10.1109/ICRIIS.2017.8002467.

SCARPELINI, P. **Estratégia para extração, transformação e armazenamento em data warehouse ativo baseada em políticas configuráveis de propagação de dados.** 2013. 65 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual Paulista, Instituto de Biociências, Letras e Ciências Exatas, São José do Rio Preto, 2013.

T. R. N; GUPTA, R. **A survey on machine learning approaches and its techniques:**. In: IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2020, Bhopal. Proceedings... Bhopal: IEEE, 2020. p. 1-6. DOI: 10.1109/SCEECS48394.2020.190.

VYAS, S.; TYAGI, R. K.; JAIN, C.; SAHU, S. **Literature review: a comparative study of real time streaming technologies and apache kafka.** In: Fourth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2021, Sonepat. Proceedings... Sonepat: IEEE, 2021. p. 146-153. DOI: 10.1109/CCICT53244.2021.00038.

WIBOWO, A. **Problems and available solutions on the stage of extract, transform, and loading in near real-time data warehousing (a literature study).** In: International Seminar on Intelligent Technology and Its Applications (ISITIA), 2015, Surabaya. Proceedings... Surabaya: IEEE, 2015. p. 345-350. DOI: 10.1109/ISITIA.2015.7220004.