ETL avançado (AETL) pela integração de PERL e método de script

Sistema de
Software Inovador Prayag Tiwari (Departamento de Ciência da Computação)
Universidade Nacional de Ciência e Tecnologia MISiS Moscou, Rússia
prayagforms@gmail.com

Resumo: Aprimorar os fluxos de dados da estrutura do processo ETL (Extração, Transformação e Carregamento) pode gerar mais lucro para o seu empreendimento comercial. Um arranjo de planejamento de nível de esforço que é tudo menos difícil de utilizar e lida com situações heterogêneas pode simplesmente fazer o que você precisa. Antes, a atenção estava voltada principalmente para o plano de processos de negócios, demonstrando. Ultimamente, os empreendimentos entenderam que podem se beneficiar imensamente da mecanização do processo de ETL com o objetivo de agilizá-los ou aprimorá-los. Com um objetivo final específico para extrair, transformar e carregar grande escala de dados de diversas fontes de dados em dados warehouse efetivamente, o AETL surgiu e foi projetado neste documento usando a sub-rotina PERL, partição de dados com integração do método de script. A principal função do AETL é aumentar a eficiência do ETL e aumentar a velocidade de processamento.

Palavras-chave: **AETL, ETL (Extrair, Transformar e Carregando), PERL, método de script**

EU. Introdução

A principal conquista para uma organização sobreviver nos últimos tempos é a capacidade de quebrar, organizar e responder às mudanças no ambiente de negócios. Essa capacidade pode ser satisfeita se dados satisfatórios estiverem acessíveis no arranjo de informações e a estrutura for apropriada com a liderança básica. Parte do desafio de programação de aplicativos hoje é fornecer dados suficientes conforme a necessidade do cliente. Melhores escolhas de chaves serão transmitidas se a natureza dos dados for sustentada por informações completas, abrangentes e precisas. Informações de baixa qualidade normalmente provocadas por um esboço de plano social terrível e ausência de requisitos rígidos de informações honestas, por exemplo, engano, repetição, irregularidade e até perda de informações.

Normalmente, a transformação e o procedimento de interface estão incluídos no processo de movimentação de dados. Ambas as estratégias assumem uma parte essencial no processo de movimentação de dados [5]. As operações médias do centro de distribuição de informações gerenciam, em grande parte, muitos Informação. O procedimento de realocação de dados está incluído no desenvolvimento de dados começando com uma estrutura e depois na próxima estrutura que inclui duas transformações, ou seja, faz uma interpretação das informações para adequar a estrutura de destino e as estratégias de interface (entrada e saída) e associa duas estruturas tendo em mente o objetivo final para sincronizar as informações.

A estrutura AETL vai para a última tabela de informações no Data Warehouse (DW) e particiona o manuseio de diversas informações em vários tipos de uso de ETL. Um trabalho ETL a longo prazo produz uma tabela de informações. O trabalho ETL, cujo último resultado será colocado na tabela de comparação no DW, é executado em uma linha de funil ETL. Eventualmente, o trabalho ETL e o pipeline ETL são aproximadamente a mesma coisa. Os diversos pipelines ETL podem ser executados em vários hosts, se essencial. A tecnologia de partição de dados é utilizada para atualizar o banco de dados, de modo a garantir o empilhamento e o questionamento proficientes das informações [10].

II. Implantação e design de AETL *A.*

Sistema ETL: ETL é a forma abreviada de extrair, transformar e carregar, três trabalhos de banco de dados que são consolidados em um sistema para transportar dados de um banco de dados e localizá-los em outro banco de dados.

Extract: A extração de dados é o procedimento de captura da fonte de dados, para a maneira de ler os dados de uma ampla gama de estruturas de operação exclusivas e purificar as informações, que é o motivo de todo o trabalho. Com a chance de não haver diretrizes de mapeamento e metadados relacionados.

Transformar: A alteração de dados é o procedimento de mudar os dados extraídos de sua estrutura passada para a estrutura com a qual deveriam estar

o objetivo que ele pode ser definido em outro banco de dados. A mudança acontece utilizando princípios ou tabelas de pesquisa ou consolidando os dados com outros dados.

Carregamento: Carregar é o procedimento de compor os dados no banco de dados objetivo. O ETL é utilizado para mover dados começando com um banco de dados e depois para o próximo, para moldar data marts e data warehouse, além de mudar os bancos de dados começando com uma organização ou classificando-os para a próxima [5].

B. *Projeto de AETL:* É o procedimento de mudança de dados da fonte para a estrutura objetiva. O trabalho ETL na estrutura AETL é um processo ETL, e a estrutura AETL é predominantemente formada pelo coletor de empregos, despachante de trabalho e pipeline ETL. nós lançar uma luz sobre a utilização de avanços de script para mecanizar dispositivos ETL preparando processo de ponta a ponta que diminui a dor cerebral manual de executar o processo ETL cuidando além disso solicita melhoria de instrumentos ETL no futuro que reforçam a interface baseada em convocação. Nós criamos a automação de um dispositivo ETL "y" que suporta "z" uma vocação usando tecnologia de script de shell. Neste artigo, estamos integrando a sub-rotina PERL, a partição de dados e o método de script que tornará o sistema ETL mais poderoso e rápido.

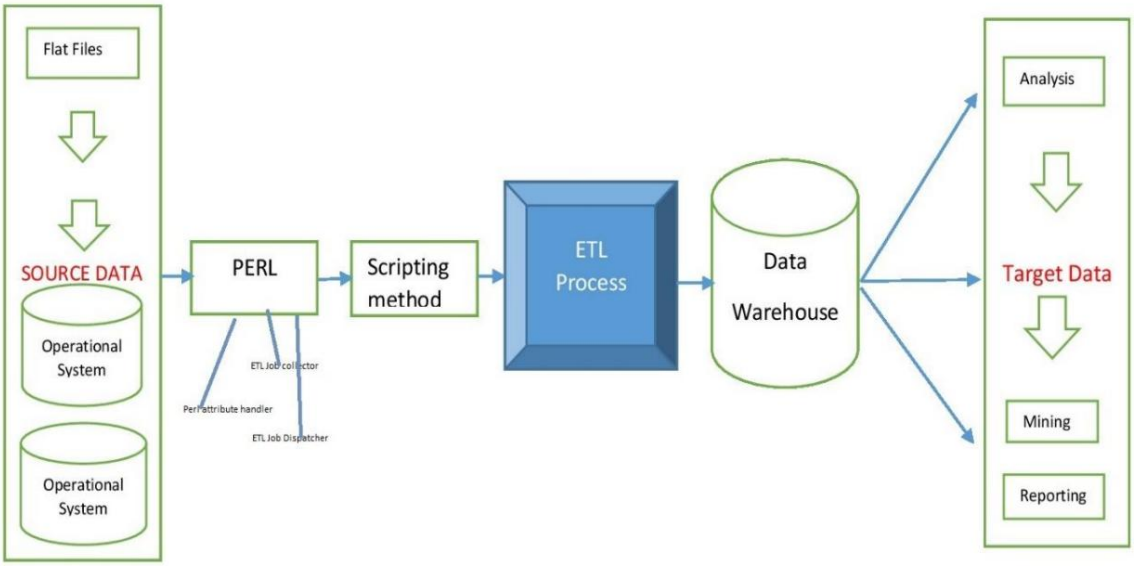

Figura 1 Projeto de AETL

O pipeline ETL é uma progressão de chamadas de sub-rotinas, e apresentaremos essas sub-rotinas na parte que o acompanha. Cada pipeline ETL pode ser executado em uma string, em um procedimento diferente ou em outro host. Esse esquema pode aumentar a produtividade da execução do ETL e não há impedância entre vários empreendimentos. Evacuar ou incluir ocupações de ETL não influenciará os atuais empregos de ETL online. Pela estrutura do módulo, o AETL possui grande adaptabilidade [11].

O despachante de trabalho é responsável por fazer pipelines ETL e despachar cada trabalho ETL para um pipeline ETL, conforme indicado pela configuração caracterizada pelo cliente. Depois que a ocupação ETL é concluída, as informações de comparação podem ser efetivamente empilhadas no banco de dados objetivo.

O coletor de tarefas é responsável por reunir as ocupações ETL caracterizadas pelos clientes, quebrando a estrutura linguística das ocupações ETL e verificando a semântica. O coletor de ocupação baseia-se na propriedade de sub-rotina do dialeto PERL.

O traço de sub-rotina do dialeto PERL pode ser ativado em meio a etapas de montagem, por exemplo, BEGIN, CHECK, INIT e END, para que a estrutura AETL possa dividir o trabalho ETL caracterizado pelo cliente no estágio de incorporação da estrutura.

C. Implantação de AETL: 1.

Sub-rotina PERL e Particionamento de Dados: Existem cinco características de sub-rotina do dialeto PERL caracterizados abaixo são utilizados para executar o coletor de tarefa ETL.

Subconfiguração: ATTR(CODE) {.....};
Subextração: ATTR(CODE){.....};
Subtransformação: ATTR(CODE){.....};
Subcarga: ATTR(CODE){.....}; Sub
desmontagem: ATTR(CODE){.....}; Existem
cinco atributos Setup, Extract, Teardown, Transform e Load
são os atributos usados durante a explicação da sub-rotina
PERL.

Subsubroutine_name: attribute_name (attribute_data)
{}

O subroutine_name é o caminho de uma sub-rotina Perl que precisa concordar com o dialeto Perl chamado tradição. O attribute_name é uma característica da sub-rotina que será recuperada no melhor passo possível no pipeline ETL. Por exemplo, o a sub-rotina com a característica "Setup" será chamada de fase inicial no pipeline ETL. O at attribute_data(ETL work name) é a estimativa da propriedade, attribute_name demonstrando que a sub-rotina tem um local com o qual o ETL trabalha, por exemplo, a sub-rotina com a característica A informação "job1" será chamada na chance de executarmos o trabalho ETL chamado "job1" no pipeline ETL. Uma sub-rotina com algum crédito pode ter um lugar com várias ocupações ETL. A razão para fazer isso é fazer com que as ocupações ETL compartilhem essas sub-rotinas. No entanto, funciona melhor no caso de uma sub-rotina ter apenas um lugar com um trabalho ETL. A classe primária de uso do AETL aparece na Figura 2.

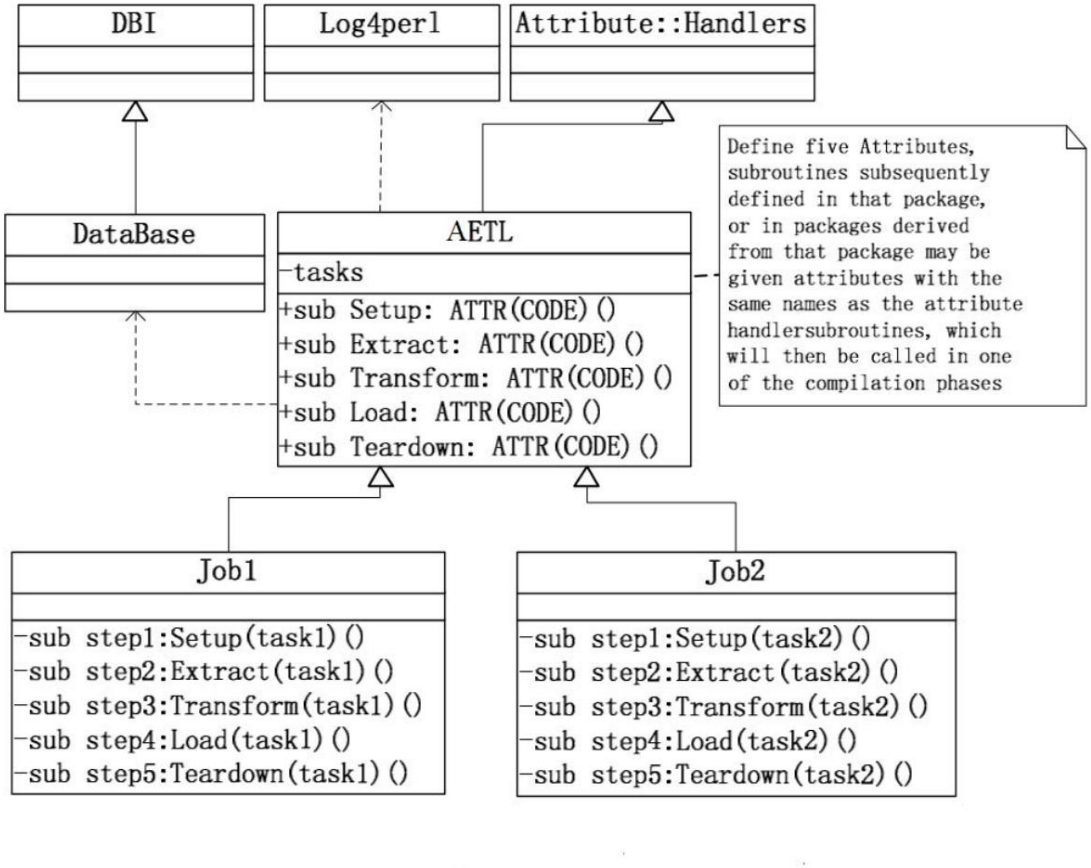

Figura 2 Sub-rotina

É demorado ao empilhar ou questionar dados enormes do banco de dados na chance de empilharmos todos os dados em uma tabela. Propomos uma técnica para melhorar a proficiência do carregamento de dados. Nesta técnica segmentamos a informação por tempo. Por exemplo, uma atribuição fala com dados de uma hora. A parcela de dados está relacionada à alocação da tabela no banco de dados. Nesta técnica, a duplicação de dados pode ser evitado durante o manuseio, o processo de carregamento de dados e a operação de consulta após o empilhamento podem ser acelerados. Considere que a baixa eficácia da explicação do SQL, AETL usa a capacidade implícita do mysql em vez da proclamação do SQL. Os testes demonstram que essa estratégia de empilhamento de informações é vinte vezes mais eficaz do que a articulação INSERT. Parte informações extensas

documentar em pequenos registros de informações é outro método para melhorar a proficiência em AETL, e os testes demonstram que

$$C \le \sum_{i1}^{n} Ti; \quad \sum_{i1}^{n}$$

Onde C é a inclusão agregada dos registros ao vivo no arquivo de dados, e ci é a contagem dos registros habitam um dos segmentos, e T é o gasto de tempo agregado de empilhamento do arquivo de dados, e o é o gasto de tempo de empilhamento um dos loteamentos. A condição acima implica que o gasto de tempo para empilhar um grande registro de uma só vez é muito maior do que todo o gasto de tempo para empilhar seus segmentos independentemente [1].

2. *Método de script:* A abordagem proposta compreende uma camada de fonte de dados que pode ter estruturas distintas homogêneas ou heterogêneas em vários hubs que podem ser **III.** registros operacionais ou de nível.

Em seguida, podemos ter a camada de parte do script, onde podemos utilizar avanços de script que podem realmente extrair as informações de uma ou mais estruturas de fonte de dados e, posteriormente, executar o processo ETL por meio de scripts pré-codificados para lidar com procedimentos específicos de extração, alteração ou carregamento. Utilizamos suportes na era de três tipos distintos de mapas e produzimos mapa de extração de origem, mapa de mudança, delineamento de empilhamento, conjurou o dispositivo ETL com a inovação de script para lidar com esses empregos de guia, além de registrar os erros e insights. A seguir está nossa proposta de cálculo de script suave que propomos para a abordagem do modelo de manipulação de data warehouse que recomendamos utilizar para preparação mecanizada que permite melhorar o processo de ETL.

Primeiro passo: Precisamos indicar as diferentes variáveis e parâmetros mundiais, entrada, caminho de saída e outros dados do ambiente, conforme exigido nos registros de projeto abaixo.

Source_path=/app/source/…
Target_path=/app/target/…
Caminho do script=/app/source/
scripts Log_path=/app/logs/
err.log
Db_name=xyz Db_password=*****

4.

Dbservername=zxy267bn

Segunda Etapa: Execute o código de script primário que convoca o sistema ETL e passe todos os parâmetros fundamentais de entrada, saída e configuração para o sistema, conforme necessário, dependendo do sistema. Para nossa situação, passamos o arquivo de alteração de mapas e o registro de configuração por meio do comando.

Terceira Etapa: Passe a posição do registro de origem, o caminho de origem e o grupo de arquivos de destino, o caminho de destino e outros dados fundamentais para o script, conjurando os registros de configuração caracterizados na primeira etapa, além de verificar se as informações existem nas fontes. Temos código de script separado para a capacidade básica de lidar com certos empreendimentos como runtime(), logingerrors(), dbcalling(), runstatus()

IV Passo: Caso vários trabalhos devam ser preparados, faça o loop deles e aplique os controles de loop.

III. Avaliação:

Depois de contemplar os dados de várias fontes, podemos inventar um modelo de como os dados se encaixam. Para capacitar isso, é necessário compreender os dados atuais e localizar as partes conectadas e, em seguida, fazer um arranjo típico para o data warehouse em um banco de dados. Com um objetivo final específico para realizar isso, podemos utilizar nossa estrutura proposta atualizada, acessível em várias estações do cliente e alterar os dados conforme o esperado para uma configuração solitária conforme necessário e armazenar em algum data warehouse onde os avanços de script assumem uma parte mais importante na preparação enormes volumes de dados de várias situações. Isso pode ser feito com sobrecarga insignificante. Qualidade inabalável é fornecida no processo de alteração e carregamento como todo o processo de carregamento com dados sendo movidos e resultados diferentes são visíveis ao cliente com o elemento fornecido em nossa estrutura. Ele fornece uma imagem clara do conhecimento de quais dados estão sendo alterados e movidos para qual estrutura de destino, etc. A exigência de configuração substancial do equipamento é dispensada no local do cliente, utilizando nossa estrutura com esforço insignificante.

4. Conclusão:

AETL torna o ETL mais rápido e aprimorado pela integração da sub-rotina PERL, partição de dados e método de script. A maioria das ferramentas atuais hoje nos negócios reforça o manuseio manual de empregos ETL. Há uma necessidade no preciso não tão

futuro distante para as ferramentas que reforçam a preparação programada de volumes de dados acessíveis e para os dispositivos ETL que suportam a interface de cliente baseada em ordem implícita para preparação mais rápida de dados e melhoria da qualidade de manipulação de dados utilizando método de script, sub-rotina PERL e aqueles que executam processamento ETL, arranjo bem como equipado para fornecer informações adicionais de registro e outros dados identificados com erros, problemas de mapeamento, tratamento de erros, número de linhas preparadas, manutenção de tabelas de revisão para resolver problemas durante a alteração.

V. Referências [1]

Vassiliadis et.al “A framework for the design of ETL Scenarios”, 15th International Conference On Advanced Information Systems Engineering, Velden, Áustria, 16 de junho de 2003.

[2] Vassiliadis, Panos; Simitsis, Akis. “Sobre a modelagem lógica de processos ETL”, Proceedings of International Conference on Advanced Information systems Engineering, 2002 pp.782-86.

[3] Alkis Simitsis, “Modelagem e gerenciamento do processo ETL”, Anais do VLDB 2003 PhD Workshop.

[4] XF Zhang, WW Sun e W. Wang, Gerando Processos ETL Incrementais Automaticamente, Ciências da Computação e da Computação, 2006, pp.516–521.

[5] H .Tahir e P. Brezillon, Uma abordagem de contexto compartilhado para apoiar especialistas em ETL de dados, processos Sistemas Inteligentes Design and Applications (ISDA 2011), IEEE Press, dezembro de 2011, pp. 720-725.

[6] C. Squire, Data Extraction and Transformation for the Data Warehouse Solutions, Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, New York:ACM,1995,pp. 446-447.

[7] A. Simitsis, Mapeamento Conceitual para Modelos Lógicos para Processos ETL, Proceedings of the 8th ACM International Workshop on Data Warehousing and OLAP. New York:ACM,2005,pp.67-76.

[8] V.Radhakrishna et.al “Implementation of Web based ETL Transformation with pré-configurado multi source system connection andtransforming mapping statistics report”, 3rd IEEE International Conference on Advanced Computer Theory and Engineering, 20-22 de agosto de 2010, Chengdu, China .

[9] Carregamento de transformação de extração - um caminho para data warehouse. 2ª Conferência Nacional Técnicas Matemáticas: Paradigmas Emergentes para as Indústrias Eletrônica e de TI. 26 a 28 de setembro de 2008.

[10] P. Vassiliadis, A. Karagiannis, V. Tziovara, P. Vassiliadis e A. Simitsis. Rumo a um benchmark para fluxos de trabalho ETL. In 5th International Workshop on Quality in Databases (QDB) at VLDB, 2007

[11] Huang Huaiyi e Yang Luming, Projeto e implementação de arquitetura leve de sistema ETL, tecnologia e desenvolvimento de computadores, vol. 18(6), junho de 2008, pp. 202-205.

[12] Zhang Zhongping e Zhao Ruizhen, Projeto de arquitetura para ETL baseado em aplicativos de computador e software controlados por metadados, vol. 26, junho de 2006, pp. 61-63,.

[13] R. Kimball, L. Reeves, M. Ross e W. Thornth-waite. O kit de ferramentas do ciclo de vida do data warehouse. John Wiley & Filhos, 1998