

MÉTODO DE MIGRAÇÃO DE TABELAS GRANDES EM BANCO DE DADOS ORACLE, PARA NUVEM COMPUTACIONAL

METHOD FOR MIGRATING LARGE TABLES IN ORACLE DATABASE TO CLOUD COMPUTING

Rubens Thiago de Oliveira¹

RESUMO

Este artigo apresenta um método sistematizado para a migração de grandes tabelas, conhecidas como "big tables", de um banco de dados Oracle, localizado em um datacenter, para uma nuvem computacional. Com o crescente aumento da demanda por soluções em nuvem, a migração de dados volumosos tornou-se um desafio crítico para as organizações, especialmente aquelas com restrições de largura de banda e orçamento. O método proposto integra ferramentas especializadas, como o Oracle Data Pump, e estratégias de particionamento de dados para assegurar a integridade e a eficiência do processo de transferência. A pesquisa detalha as etapas de planejamento, execução e validação da migração, fornecendo um guia prático para profissionais de TI. Os resultados demonstram a viabilidade do método, destacando tanto as vantagens quanto as limitações identificadas. Este estudo contribui para o campo ao fornecer uma solução robusta para a migração de dados em larga escala, com implicações práticas para a adoção de tecnologias de computação em nuvem.

Palavras-chave: Migração de Banco de Dados; Tabelas Grandes; Computação em Nuvem; Tabelas Externas; Otimização de Transferência de dados.

ABSTRACT

This article presents a systematic method for migrating large tables, known as "big tables," from an Oracle database located in a datacenter to a cloud computing environment. With the increasing demand for cloud solutions, the migration of large volumes of data has become a critical challenge for organizations, particularly those with bandwidth and budget constraints. The proposed method integrates specialized tools, such as Oracle Data Pump, and data partitioning strategies to ensure the integrity and efficiency of the transfer process. The research

1. Mestre em ciência da computação, arquiteto, cientista de dados, professor de tecnologia da informação da Faculdade das Américas (FAM) e professor convidado da Universidade Estadual de Campinas (UNICAMP) – Contato: rubenst@m.unicamp.br. ORCID: [\[0009-0006-4794-4350\]](https://orcid.org/0009-0006-4794-4350)

details the stages of planning, execution, and validation of the migration, providing a practical guide for IT professionals. The results demonstrate the feasibility of the method, highlighting both the identified advantages and limitations. This study contributes to the field by offering a robust solution for large-scale data migration, with practical implications for the adoption of cloud computing technologies.

Keywords: Database Migration; Big tables; Cloud Computing; External Tables; Data Transfer Optimization.

1. INTRODUÇÃO

Com o surgimento crescente de novos ambientes computacionais, as empresas enfrentam desafios crescentes na migração de suas bases de dados, especialmente aquelas que possuem grandes volumes de dados (FAHMIDEH et al., 2017; FISHER, 2024). A migração para a nuvem tornou-se uma estratégia crucial no cenário de TI, impulsionada pela escalabilidade, flexibilidade e redução de custos operacionais (ELLISON et al., 2018). Segundo (DZIUBAK, 2023), a computação em nuvem permite que as empresas aloquem recursos de forma dinâmica, eliminando a necessidade de grandes investimentos em infraestrutura física.

Durante o processo de migração de grandes ambientes, principalmente para a nuvem computacional, é frequentemente necessário contratar capacidades adequadas de serviço de link de rede entre o datacenter e o provedor de nuvem (FAHMIDEH et al., 2017). A seguir, é apresentado um estudo sobre as taxas de transferência entre o datacenter e o fornecedor de nuvem computacional (AWS), uma escolha estratégica devido à sua liderança de mercado e à robustez dos serviços oferecidos.

TRANSFERÊNCIAS ENTRE DATACENTERS ➡ AWS										
Link (Mbps)	Taxa de Transferência (MB/s)	TEMPO								
		100% de uso da banda			70% de uso da banda			40% de uso da banda		
		(min)	(h)	(dias)	(min)	(h)	(dias)	(min)	(h)	(dias)
1	0,13	13.653	228	9	17.749	296	12	21.845	364	15
50	6,25	273	5	0	355	6	0	437	7	0
100	12,50	137	2	0	177	3	0	218	4	0
200	25	68	1	0	89	1	0	109	2	0
300	37,50	46	1	0	59	1	0	73	1	0
400	50	34	1	0	44	1	0	55	1	0
500	62,50	27	0	0	35	1	0	44	1	0

Figura 1: Simulação das taxas de transferências entre o datacenter e a AWS.
Fonte: Elaborado pelo autor

A Figura 1 apresenta um estudo sobre os tempos das taxas de transferência de um arquivo com o tamanho de aproximadamente 100 GB. Os tempos de transferência estão expressos em minutos, horas e dias, variando de acordo com a taxa de transferência contratada. Para esta simulação, considerou-se que o link opera, em sua maioria, com 40% (quarenta por cento) de uso da banda; contudo, caso operasse com um uso superior (>40%), os tempos de transferência poderiam apresentar melhor eficiência.

Esta análise pode se mostrar como um fator crítico durante o processo de migração, pois, em muitas situações, os tempos de transferência podem dilatar as janelas de manutenção necessárias para a execução da migração, inviabilizando os trabalhos. Em cenários em que as empresas enfrentam restrições orçamentárias que dificultam a contratação de links de grande capacidade ou serviços de transporte de dados, torna-se essencial recorrer a mecanismos alternativos para a realização da migração de dados de maneira eficiente e segura (HUSSEIN, 2021).

A escolha da Oracle como banco de dados e da AWS como provedor de nuvem fundamenta-se na necessidade de adotar tecnologias consolidadas que ofereçam suporte robusto e flexível, atendendo às demandas dos ambientes empresariais contemporâneos. Essa combinação facilita a migração eficiente dos dados e permite que as empresas aproveitem benefícios da nuvem, como escalabilidade sob demanda e alta disponibilidade, aspectos essenciais no contexto competitivo atual.

1.1. OBJETIVO

O objetivo deste artigo é desenvolver um procedimento para a migração eficiente de grandes tabelas, conhecidas como "big tables", utilizando ferramentas especializadas para transferência de dados entre ambientes on-premises (datacenters locais) e plataformas de computação em nuvem (cloud computing) (ELLISON et al., 2018; FISHER, 2024). Essas informações podem beneficiar empresas e analistas de banco de dados, fornecendo orientações claras e práticas para a execução bem-sucedida da migração de dados.

Embora o método seja delineado com foco nas ferramentas descritas, ele pode ser adaptado para outros contextos que envolvam diferentes tecnologias de banco de dados e plataformas de nuvem, mediante ajustes apropriados nas etapas do processo (HUSSEIN, 2021).

1.2. MOTIVAÇÃO

A motivação para esta pesquisa baseia-se na necessidade de desenvolver procedimentos criteriosos e bem estruturados para conduzir a migração de tabelas com grandes volumes de dados de maneira eficaz e organizada. Diretrizes claras e detalhadas têm o potencial de fornecer às empresas orientações precisas, facilitando uma migração segura e eficiente de seus bancos de dados.

1.3. MATERIAIS

Este estudo foi realizado utilizando as seguintes ferramentas técnicas:

- Banco de dados: Oracle 12c para armazenamento e gestão das tabelas do sistema SAP/R3;
- Ferramenta de transferência de dados: Oracle Data Pump, para exportação e importação de grandes volumes de dados;
- Plataforma de computação em nuvem: AWS Cloud Computing, utilizada para hospedar o ambiente de destino dos dados migrados;
- Armazenamento em nuvem: Amazon S3, para armazenamento temporário de dados exportados e transferidos do ambiente de on-premises.

1.4. METODOLOGIA DE TRABALHO

As etapas deste trabalho foram desenvolvidas para demonstrar o processo de migração de uma tabela do banco de dados SAP/R3, de um ambiente de datacenter para a nuvem computacional. As etapas seguidas foram:

1. Planejamento da exportação da tabela do banco de dados no datacenter para a nuvem;
2. Desenvolvimento dos scripts de exportação de dados no datacenter;
3. Transferência dos dados exportados para a nuvem computacional;
4. Criação da estrutura da tabela no banco de dados na nuvem;
5. Importação dos dados transferidos para o novo ambiente;
6. Validação dos dados na tabela migrada.

2. REVISÃO BIBLIOGRÁFICA

2.1. ORACLE DATABASE ENTERPRISE 12C

O Oracle Enterprise 12c é uma plataforma de gerenciamento de bancos de dados amplamente reconhecida por sua escalabilidade, segurança e desempenho (ORACLE, 2024). Destaca-se por seus recursos avançados de computação em nuvem, como multitenancy e automação, que permitem um gerenciamento eficiente e flexível de ambientes empresariais críticos.

2.2. SAP/R3

O SAP R/3 é uma versão proeminente do sistema de gestão empresarial (ERP) lançado em 1992, destacando-se por introduzir uma arquitetura cliente/servidor e uma interface gráfica, em substituição aos mainframes centralizados. Reconhecido por sua modularidade e escalabilidade, o SAP R/3 abrange áreas funcionais como finanças, contabilidade e recursos humanos, e teve grande influência na disseminação global de sistemas ERP (ORACLE, 2024).

2.3. ORACLE DATA PUMP

O Oracle Data Pump é uma ferramenta avançada de transferência de dados (ORACLE, 2024), altamente eficiente e confiável, amplamente utilizada em ambientes de bancos de dados Oracle para importar e exportar grandes volumes de dados. Oferece recursos como compressão, paralelismo e filtragem, sendo aplicado em cenários que exigem movimentação de dados em larga escala, como a realização de backups lógicos e a execução de migrações entre ambientes de bancos de dados.

2.4. AWS CLOUD COMPUTING

A AWS Cloud Computing é uma plataforma de computação em nuvem fornecida pela Amazon Web Services, oferecendo uma ampla gama de serviços de infraestrutura, armazenamento e computação, disponibilizados de forma escalável e sob demanda (SAMPATHKUMAR e KATARI, 2024). Caracterizada por sua flexibilidade, confiabilidade e segurança, a AWS é amplamente utilizada por organizações para hospedar aplicativos, armazenar dados e executar cargas de trabalho diversas na nuvem. Seus serviços abrangentes abrem oportunidades para inovação e transformação digital em uma variedade de setores e aplicações.

2.5. AWS S3

O Amazon S3 (Simple Storage Service) é um serviço de armazenamento em nuvem da Amazon Web Services (AWS), altamente escalável, seguro e durável, projetado para armazenar e recuperar objetos digitais de forma confiável e acessível (ZHAO e ZHOU, 2015). É amplamente utilizado em diversas aplicações devido à sua flexibilidade, baixa latência e facilidade de integração.

2.5. DISCUSSÃO COMPARATIVA

Enquanto o Oracle Data Pump se destaca pela eficiência em ambientes Oracle, outras

ferramentas de migração (ORACLE, 2024), como o AWS Database Migration Service (DMS), oferecem flexibilidade para migrações entre diferentes plataformas de banco de dados e ambientes híbridos (SAMPATHKUMAR e KATARI, 2024). A AWS Cloud Computing proporciona uma gama abrangente de serviços que podem influenciar a escolha da solução de migração mais adequada para diferentes necessidades empresariais. Ao integrar esses elementos, o estudo evidencia que a escolha de ferramentas e plataformas deve ser orientada pelas especificidades do ambiente de migração e pelos requisitos de desempenho e escalabilidade das operações.

3. DESENVOLVIMENTO DO TRABALHO

Nesta seção, são detalhadas as etapas de desenvolvimento do trabalho, abrangendo desde o planejamento da migração da tabela, a definição dos procedimentos de migração, a transferência dos dados para o novo ambiente e a restauração das tabelas.

3.1 PLANEJAMENTO DA MIGRAÇÃO DA TABELA

A tabela escolhida para este estudo foi a KOCLU, que é a maior tabela do sistema SAP/R3, com tamanho de 679 GB aproximadamente e com o total de 101.302.440 milhões de linhas. A seguir é apresentada as características desta tabela.

TABELA - SAPSR3.KOCLU

Name	Null?	Type
MANDT	NOT NULL	VARCHAR2(9)
KNUMV	NOT NULL	VARCHAR2(30)
PAGENO	NOT NULL	NUMBER(5)
TIMESTAMP	NOT NULL	VARCHAR2(138)
PAGELG	NOT NULL	NUMBER(5)
VARDATA		BLOB

Figura 2: Tabela KOCLU.

Fonte: Elaborado pelo autor.

As características da tabela KOCLU, apresentada na Figura 2, permitem observar a definição de 5 (cinco) colunas, que não aceitam valores nulos, e estão utilizando tipos de dados VARCHAR2 e NUMBER e uma coluna utilizando o tipo de dados BLOB.

O tipo de dados BLOB, é utilizado para armazenar grandes volumes de dados binários, como imagens, vídeos, documentos, entre outros tipos de informação. É especialmente útil quando se deseja armazenar informações não estruturadas ou de grande tamanho, que não podem ser facilmente representadas em formato de texto. O tipo de dado BLOB armazena os dados binários exatamente como foram inseridos, preservando sua integridade e permitindo sua recuperação posterior de forma precisa.

Durante os testes preliminares de migração da tabela mencionada, ocorreram erros, que impediram a realização da exportação total de dados da tabela. A seguir estão apresentados os erros encontrados durante os testes preliminares.

```
ORA-31693: Table data object "SAPSR3"."KOCLU" failed to  
          load/unload and is being skipped due to error:  
ORA-02354: error in exporting/importing data  
ORA-01555: snapshot too old: rollback segment number 12  
          with name "_SYSSMU12_1378260260$" too small
```

Figura 3: Erros durante o processo de migração.

Fonte: Elaborado pelo autor.

Durante a operação de exportação dos dados completos da tabela, a operação foi cancelada devido ao erro, apresentado na figura 3. Este erro está relacionado ao volume de dados da tabela juntamente com a coluna de tipo binário. Como solução, para evitar os erros durante a migração, foi realizada uma análise quantitativa na tabela KOCLU, para permitir que a tabela fosse dividida em partes iguais.

O após estudos a tabela KOCLU, que possuía 101.302.445 milhões de linhas foi dividida em oito partes iguais contendo 12.662.806 milhões de linhas. Para executarmos esta análise, foi executado o comando apresentado abaixo.

```
select min(rowid), max(rowid), tile from (select /*+ parallel (a,40) */ rowid,
ntile(8) over (order by rowid) as tile from sapsr3.koclu a) group by tile order
by 1 ;
```

O resultado deste comando foi a divisão da tabela nas seguintes partes.

MIN(ROWID)	MAX(ROWID)	TILE
-----	-----	----
AAAs/sAAEAABHyCAAA	AAAs/sAA3AAM76HAAA	1
AAAs/sAA3AAM76IAAA	AAAs/sABNAAHte1AAA	2
AAAs/sABNAAHtemAAA	AAAs/sABhAAK7xeAAA	3
AAAs/sABhAAK7xfAAA	AAAs/sAB4AAJY8DAAA	4
AAAs/sAB4AAJY8EAAA	AAAs/sACMAAF2xfAAB	5
AAAs/sACMAAF2xfAAC	AAAs/sACkAAP+KUAAD	6
AAAs/sACkAAP+KUAAE	AAAs/sADMAANO+IAAA	7
AAAs/sADMAANO+JAAA	AAAs/sAD+AAEh9/AAA	8

8 rows selected.

Figura 4: Resultado da divisão da tabela KOCLU em 8 partes.

Fonte: Elaborado pelo autor.

A divisão da tabela, apresentada na Figura 4, representam o intervalo inicial e final de cada registro, respectivamente, indicados pelos valores das funções MIN(ROWID) e MAX(ROWID). Cada parte desta divisão apresentou uma quantidade de 12.662.805 milhões de linhas. O objetivo da divisão da tabela KOCLU, foi permitir o envio da tabela para o novo ambiente computacional, utilizando a infraestrutura disponibilizada para o estudo.

3.2. DEFINIÇÃO DA MIGRAÇÃO DA TABELA

Para a exportação dos dados, foram definidos o uso de tabelas externas, pertinente a cada divisão criada no planejamento, conforme apresentado na figura 5.

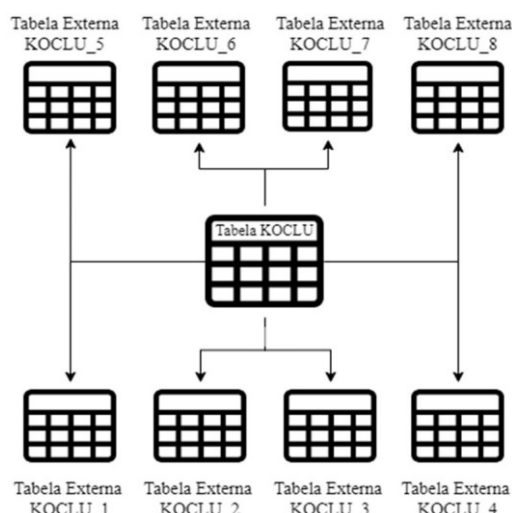


Figura 5: Diagrama representativo da exportação dos dados em 8 partes.

Fonte: Elaborado pelo autor.

Para a criação das tabelas externas foram utilizados os seguintes comandos:

```
create table sapsr3.koclu_external_1
organization external(
  type oracle_datapump default directory expdp
  access parameters (logfile 'koclu_external_1.log')
  location ('koclu_external_1_01.dmp'))
parallel 40
as
select /*+ parallel (a,40) */ *
  from sapsr3.koclu a
 where rowid between charrowid('AAAs/sAAEAABHyCAAA')
    and charrowid('AAAs/sAA3AAM76HAAA');
```

Figura 6: Comando de criação da tabela externa para exportação do dado.

Fonte: Elaborado pelo autor.

Observando o script de criação da tabela externa, apresentada na Figura 6, é possível verificar a chamada da ferramenta Oracle Data Pump, que é utilizado como saída de um arquivo, denominado “koclu_external_1_01.dmp”, que possui o formato de um arquivo de “dump”, a mesma saída utilizada para a exportação de dados. Para acelerar o processamento de exportação da tabela, esta foi paralelizada em 40 processos.

```
organization external(
  type oracle_datapump default directory expdp
  access parameters (logfile 'koclu_external_1.log')
  location ('koclu_external_1_01.dmp'))
```

Figura 7: Detalhe da configuração da chamada do Oracle Data Pump, junto a criação da tabela externa.
Fonte: Elaborado pelo autor.

Na Figura 8, também é possível verificar como foi criada a divisão da tabela, que está referenciada pelo comando.

```
select /*+ parallel (a,40) */ *
  from sapsr3.koclu a
 where rowid between chartorowid('AAAs/sAAEAABHyCAAA')
    and chartorowid('AAAs/sAA3AAM76HAAA');
```

Figura 8: Comando que permitiu a criação da 1ª divisão da tabela.
Fonte: Elaborado pelo autor.

Todo o ciclo de criação das tabelas externas, foram realizadas a partir das variações nos nomes das tabelas, dos nomes dos arquivos de “dump” e dos gerados dos intervalos das divisões, respectivamente, representados nas Figuras 9, 10 e 11.

```
create table sapsr3.koclu_external_1...
create table sapsr3.koclu_external_2...
create table sapsr3.koclu_external_3...
create table sapsr3.koclu_external_4...
create table sapsr3.koclu_external_5...
create table sapsr3.koclu_external_6...
create table sapsr3.koclu_external_7...
create table sapsr3.koclu_external_8...
```

Figura 9: Definição dos nomes utilizados para as tabelas externas.
Fonte: Elaborado pelo autor.

```

koclu_external_1_01.dmp
koclu_external_2_01.dmp
koclu_external_3_01.dmp
koclu_external_4_01.dmp
koclu_external_5_01.dmp
koclu_external_6_01.dmp
koclu_external_7_01.dmp
koclu_external_8_01.dmp

```

Figura 10: Definição dos nomes utilizados para os arquivos de “dump”.
Fonte: Elaborado pelo autor.

```

rowid between charrowid('AAAs/sAAEAABHyCAAA') and charrowid('AAAs/sAA3AAM76HAAA')
rowid between charrowid('AAAs/sAA3AAM76IAAA') and charrowid('AAAs/sABNAAHtelAAA')
rowid between charrowid('AAAs/sABNAAHtemAAA') and charrowid('AAAs/sABhAAK7xeAAA')
rowid between charrowid('AAAs/sABhAAK7xfAAA') and charrowid('AAAs/sAB4AAJY8DAAA')
rowid between charrowid('AAAs/sAB4AAJY8EAAA') and charrowid('AAAs/sACMAAF2xfAAB')
rowid between charrowid('AAAs/sACMAAF2xfAAC') and charrowid('AAAs/sACKAAP+KUAAD')
rowid between charrowid('AAAs/sACKAAP+KUAAE') and charrowid('AAAs/sADMAANO+IAAA')
rowid between charrowid('AAAs/sADMAANO+JAAA') and charrowid('AAAs/sAD+AAEH9/AAA')

```

Figura 11: Intervalos utilizados para a divisão da tabela em oito partes.
Fonte: Elaborado pelo autor.

A cada finalização de criação de uma tabela externa, uma nova tabela foi criada, até que a finalização das oito tabelas fosse concluída.

O propósito deste procedimento é assegurar que cada tabela externa seja exportada do banco de dados do data center para a nuvem computacional, a fim de controlar as quantidades de registros migrados, bem como otimizar a utilização do link de transferência entre os ambientes on-premises e da nuvem computacional.

3.3. TRANSFERÊNCIA DOS DADOS DO DATACENTER PARA A NUVEM COMPUTACIONAL

Observando a figura 12, é possível, compreender os passos necessários para a realização da transferência dos dados entre o datacenter e a nuvem computacional.

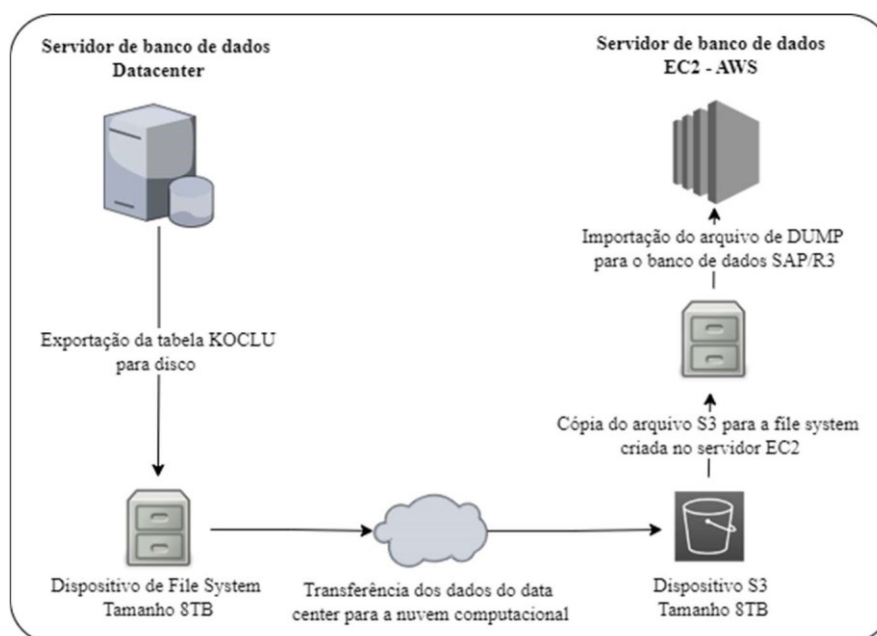


Figura 12: Desenho da arquitetura utilizada para a migração dos dados da tabela KOCLU.

Fonte: Elaborado pelo autor.

Inicialmente foi realizada a exportação dos dados da tabela KOCLU do banco de dados para os discos adicionais que foram atachados no servidor – para armazenar os dados exportados da tabela, foi disponibilizado uma capacidade de armazenamento de 8 (oito) terabytes, capacidade superior ao tamanho da tabela.

A etapa subsequente envolveu a transferência do arquivo gerado do file system, localizado no datacenter para o dispositivo S3, localizado na nuvem computacional. Na sequência foi executado uma cópia do arquivo do dispositivo S3 para a file system, que foi disponibilizada no servidor EC2.

Após a conclusão da transferência do dado do datacenter para a nuvem computacional, o processo de importação dos dados para a tabela pode ser iniciado.

Este processo foi executado serialmente por 8 (oito) vezes, referente as 8 divisões da tabela exportada, o objetivo desta prática foi de otimizar a utilização do link disponibilizado para a transferência dos dados.

3.4. IMPORTAÇÃO DOS DADOS TRANSFERIDOS DO DATACENTER PARA A NUVEM COMPUTACIONAL

O passo seguinte, após a transferência do primeiro arquivo para a nuvem computacional, foi a criação da tabela externa que recebeu as informações.

Para o processo de criação foram criadas 8 tabelas externas que leram os dados exportados em arquivos “dump” e gravaram em tabelas dentro do banco de dados, a seguir é apresentado o script da 1ª. parte da tabela a ser importada na base de dados.

```
create table sapsr3.koclu_external_1
(MANDT    VARCHAR2(9),
 KNUMV    VARCHAR2(30),
 PAGENO    NUMBER(5),
 TIMESTMP VARCHAR2(138),
 PAGELG    NUMBER(5),
 VARDATA   BLOB)
  organization external (
    type oracle_datapump default directory impdp
    access parameters (logfile 'koclu_external_1.log')
    location ('koclu_external_1_01.dmp'));
```

Figura 13: Comando de criação da tabela externa para importação do dado.
Fonte: Elaborado pelo autor.

Observando o script de criação da tabela externa, apresentada na Figura 14, é possível verificar a chamada da ferramenta Oracle Data Pump, que é utilizado como saída de um arquivo, denominado “koclu_external_1_01.dmp”, que possui o formato de um arquivo de “dump”.

```
organization external (
  type oracle_datapump default directory impdp
  access parameters (logfile 'koclu_external_1.log')
  location ('koclu_external_1_01.dmp'));
```

Figura 14: Detalhe da configuração da chamada do Oracle Data Pump, junto a criação da tabela externa.
Fonte: Elaborado pelo autor

Para as demais criações das tabelas externas, foram realizadas variações nos nomes das

tabelas, nomes dos arquivos de dump gerados dos intervalos das divisões, respectivamente, apresentados nas Figuras 15 e 16.

```
create table sapsr3.koclu_external_1...  
create table sapsr3.koclu_external_2...  
create table sapsr3.koclu_external_3...  
create table sapsr3.koclu_external_4...  
create table sapsr3.koclu_external_5...  
create table sapsr3.koclu_external_6...  
create table sapsr3.koclu_external_7...  
create table sapsr3.koclu_external_8...
```

Figura 15: Definição dos nomes utilizados para as tabelas externas.

Fonte: Elaborado pelo autor.

```
koclu_external_1_01.dmp  
koclu_external_2_01.dmp  
koclu_external_3_01.dmp  
koclu_external_4_01.dmp  
koclu_external_5_01.dmp  
koclu_external_6_01.dmp  
koclu_external_7_01.dmp  
koclu_external_8_01.dmp
```

Figura 16: Definição dos nomes utilizados para os arquivos de “dump”.

Fonte: Elaborado pelo autor.

3.5. CONTAGEM DAS LINHAS MIGRADAS DA TABELA

Após a criação das tabelas externas, cada parte teve suas linhas contadas antes da importação final para a tabela original, o objetivo da contagem foi garantir que as quantidades de linhas exportadas seriam mantidas durante o processo de migração. O comando a seguir, apresenta como foi realizada a contagem.

```

select /*+ parallel(a,40) */ count(*) from SAPSR3.koclu_external_1 a; select /*+
parallel(a,40) */ count(*) from SAPSR3.koclu_external_2 a; select /*+ parallel(a,40) */
count(*) from SAPSR3.koclu_external_3 a; select /*+ parallel(a,40) */ count(*) from
SAPSR3.koclu_external_4 a; select /*+ parallel(a,40) */ count(*) from
SAPSR3.koclu_external_5 a; select /*+ parallel(a,40) */ count(*) from
SAPSR3.koclu_external_6 a; select /*+ parallel(a,40) */ count(*) from
SAPSR3.koclu_external_7 a; select /*+ parallel(a,40) */ count(*) from
SAPSR3.koclu_external_8 a;

```

3.6. CARGA DAS TABELAS TEMPORÁRIAS PARA A TABELA PRINCIPAL DO BANCO DE DADOS

Após a contagem das tabelas externas, foi a vez da concentração de todos os dados das tabelas temporárias externas para tabela principal do banco de dados, conforme esquema apresentado na figura 17.

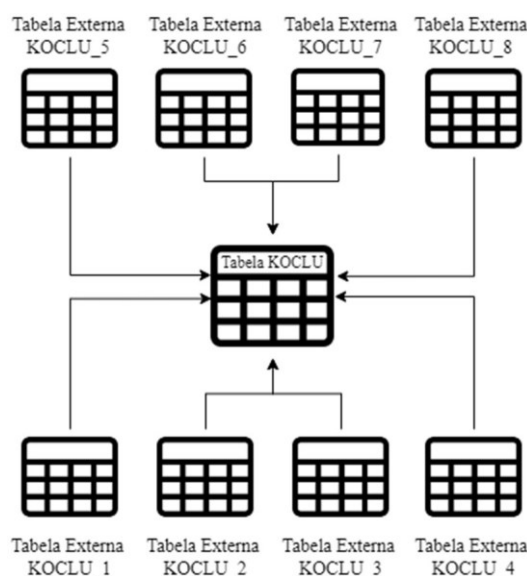


Figura 17: Diagrama representativo da integração dos dados das tabelas temporárias para a tabela principal do sistema. Fonte: Elaborado pelo autor.

Para a execução, desta da integração dos dados, foi executado o comando a seguir.

```
INSERT INTO SAPSR3.KOCLU  
(MANDT,KNUMV,PAGENO,TIMESTAMP,PAGELG,VARDATA)  
SELECT /*+ PARALLEL (A,40) */  
MANDT,KNUMV,PAGENO,TIMESTAMP,PAGELG,VARDATA  
FROM SAPSR3.KOCLU_EXTERNAL_1 A ;
```

Para cada integração da tabela externa foi executado o comando acima, com as variações dos nomes das tabelas externas, conforme apresentado a seguir.

```
FROM SAPSR3.KOCLU_EXTERNAL_1  
FROM SAPSR3.KOCLU_EXTERNAL_2  
FROM SAPSR3.KOCLU_EXTERNAL_3  
FROM SAPSR3.KOCLU_EXTERNAL_4  
FROM SAPSR3.KOCLU_EXTERNAL_5  
FROM SAPSR3.KOCLU_EXTERNAL_6  
FROM SAPSR3.KOCLU_EXTERNAL_7  
FROM SAPSR3.KOCLU_EXTERNAL_8
```

Após a finalização da integração das tabelas externas, foi realizado uma nova contagem geral da tabela KOCLU, o objetivo era garantir que nenhum registro foi rejeitado durante a integração da tabela original. Para a contagem geral dos registros foi utilizado o comando abaixo.

```
select /*+ parallel (a,40) */ count(*) from SAPSR3.KOCLU a ;
```

4. RESULTADOS

Nesta seção, são apresentados os resultados obtidos durante a execução do método proposto para a migração da tabela KOCLU. O foco principal está no tempo total de execução da atividade e no impacto das variáveis relacionadas ao link de rede utilizado durante a transferência dos dados.

4.1. TEMPO DE EXECUÇÃO TOTAL DA ATIVIDADE

O tempo total necessário para completar a migração da tabela KOCLU, com aproximadamente 101 milhões de linhas, foi de 43 dias. Este tempo foi diretamente influenciado pela capacidade do link de rede, que operava com uma taxa de utilização de throughput de aproximadamente 25%, dado que o link não era de uso exclusivo para o estudo.

A Figura 18 ilustra o cálculo do tempo de execução total, destacando a relação entre o tamanho do link e o tempo necessário para concluir a migração.

TEMPO EXECUCAO			
Migração Tabela KOCLU - SAP/R3			
	TEMPO	TAMANHO em GB	
Início	28/10/23 10:00	ATUAL	FINAL
Final	4/11/23 18:00	1.036	6.123
Total	176:00:00	10.560	X
X =		62.384	Minutos
		1.040	Horas
		43	Dias

Figura 18: Tempo de execução da atividade.
Fonte: Elaborado pelo autor.

Durante a pesquisa, o link de 500K utilizado mostrou-se insuficiente para uma migração rápida, resultando em um prolongamento significativo do processo.

Link utilizado durante a pesquisa

Link (Mbps)	(*) Taxa de Transferência (MB/s)	Tamanho Arquivo (MB)	Tempo		
			(min)	(h)	(dias)
0,5	0,06	102.400	62.123	1.035	43

(*) 25% de uso da banda

Figura 19: Link utilizado durante a pesquisa.
Fonte: Elaborado pelo autor.

A Figura 19 detalha a configuração do link utilizado e sua capacidade de throughput durante o estudo, ressaltando as limitações impostas pela largura de banda disponível.

Para avaliar o impacto de diferentes capacidades de link na eficiência da migração, foi realizada uma simulação que pode ser observada na Figura 20, que projetou os tempos de execução em função de links de maior capacidade.

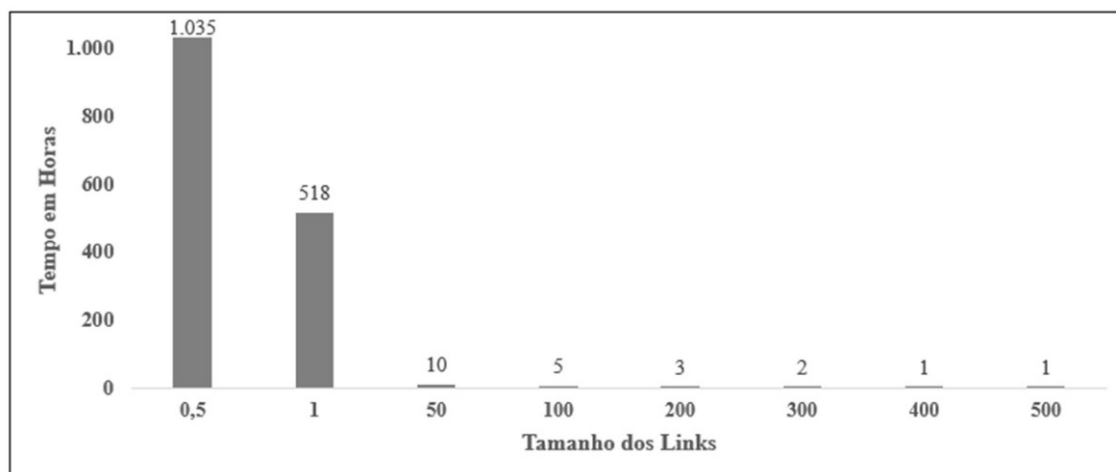


Figura 20: Simulação dos tempos em horas de execução x a capacidade do link.
Fonte: Elaborado pelo autor.

Os resultados indicam que, com um link de 50 Mbps, o tempo de execução poderia ser reduzido de 62.123 minutos para 311 minutos, conforme apresentado na Figura 21. Essa

redução significativa no tempo evidencia a importância de um link de alta capacidade para otimizar o processo de migração de grandes volumes de dados.

Link utilizado durante a pesquisa			
Links (Mbps)	25% de uso da banda		
	Tempo (min)	Tempo (h)	Tempo (dias)
0,5	62.123	1.035	43
1	31.061	518	22
50	621	10	0,4
100	311	5	0,2
200	155	3	0,1
300	104	2	0,1
400	78	1	0,1
500	62	1	0,0

Figura 21: Relação de links x tempo de execução do processo de migração.
Fonte: Elaborado pelo autor.

5. CONCLUSÃO

O método proposto para a migração de tabelas grandes em banco de dados Oracle para a nuvem computacional demonstrou ser eficaz, especialmente em ambientes onde os dados são estáticos ou legados.

As etapas detalhadas garantem que qualquer tamanho de tabela pode ser migrado para a nuvem, desde que as considerações de infraestrutura e link de rede sejam adequadamente gerenciadas.

No entanto, a pesquisa identificou limitações, como a dependência de um link de alta capacidade e a necessidade de execução serial dos scripts, o que sugere áreas para melhorias, como a aplicação de compressão de dados para acelerar a transferência.

Em ambientes de evolução contínua, a criação de uma janela de manutenção torna-se essencial para evitar interrupções (VODOMIN; ANDROČEC, 2015). Futuros estudos podem explorar a automatização do processo de migração e sua adaptação para diferentes ambientes

de nuvem, ampliando a aplicabilidade e eficiência do método.

5.1. DIFICULDADES ENCONTRADAS

A pesquisa enfrentou diversas dificuldades que influenciaram diretamente a eficácia do processo de migração.

A limitação do link de rede, operando com apenas 25% de utilização do throughput disponível, resultou em tempos de migração prolongados (BARKER et al., 2012; HUSSEIN, 2024). Para mitigar essa limitação, a pesquisa sugere a consideração de técnicas de compressão de dados e o uso de links de maior capacidade em futuros estudos.

A necessidade de execução serial dos scripts também prolongou o processo, destacando a importância de explorar técnicas de paralelização mais avançadas para otimizar o tempo de execução. Além disso, ajustes nos tamanhos dos arquivos gerados foram necessários para permitir uma cópia mais eficaz dos dados, sugerindo que a personalização dos parâmetros de exportação é crucial em cenários de migração complexos.

Essas dificuldades indicam áreas para melhorias futuras particularmente em ambientes com diferentes recursos de rede ou utilizando outras plataformas de nuvem.

5.2. APLICABILIDADE DA PESQUISA

O método proposto demonstra sua eficácia em ambientes legados, onde os dados são relativamente estáticos, oferecendo uma solução prática para a migração de grandes tabelas para a nuvem computacional. Contudo, em cenários de evolução contínua, onde os dados estão em constante atualização, o método enfrenta limitações significativas.

Para esses casos, a criação de uma janela de manutenção programada é essencial, embora isso possa não ser viável em todos os ambientes de produção. Futuros estudos poderiam explorar a integração deste método com técnicas de replicação em tempo real ou sincronização contínua de dados, ampliando sua aplicabilidade em ambientes que exigem alta disponibilidade.

Além disso, o método pode ser adaptado para ambientes híbridos, combinando a migração para a nuvem com a manutenção de uma infraestrutura on-premises, proporcionando maior flexibilidade e resiliência às organizações.

Essa versatilidade sugere que, com adaptações apropriadas, o método pode ser uma ferramenta valiosa em uma ampla gama de cenários de migração.

6. REFERÊNCIAS

BARKER, S.; CHI, Y.; MOON, H. J.; HACIGÜMÜS; SHENOY, P. Cut me some slack: Latency-aware live migration for databases. *Proceedings of the 15th International Conference on Extending Database Technology*, p. 432-443, ACM, 2012.

DZIUBAK, S. Review of Cloud Database Benefits and Challenges. *Modern Management Review*, Polônia, v. 28, n. 3, p. 7-16, jul.-set. 2023.

ELLISON, M.; CALINESCU, R.; PAIGE, R. Evaluating cloud database migration options using workload models. *Journal of Cloud Computing: Advances Systems and Applications*, v. 6, 2018. ISSN 2192-113X.

FAHMIDEH, M.; DANESHGAR, F.; BEYDOUN, G.; RABHI, F. Challenges in migrating legacy software systems to the cloud—an empirical study. *Information Systems*, v. 67, p. 100-113, jul. 2017.

FISHER, C. Cloud versus On-Premise Computing. *American Journal of Industrial and Business Management*, v. 8, p. 1991-2006, 2018. ISSN Online: 2164-5175. ISSN Print: 2164-5167. Disponível em: <http://www.scirp.org/journal/ajibm>. Acesso em: 12 ago. 2024.

HUSSEIN, A. A. Data Migration Need Strategy Challenges Methodology Categories Risks Uses with Cloud Computing and Improvements in Its Using with Cloud Using Suggested Proposed Model (DMig 1). *Journal of Information Security*, v. 12, n. 2, p. 79-103, 2021. ISSN Online: 2153-1242. ISSN Print: 2153-1234. Disponível em: <https://www.scirp.org/journal/jis>. Acesso em: 28 ago. 2024.

JAMSHIDI, P.; AHMAD, A.; PAHL, C. Cloud Migration Research: A Systematic Review. *IEEE Transactions on Cloud Computing*, v. 1, p. 142-157, out. 2013.

KALUBA, Z.; NYIRENDA, M. Database Migration Service with A Microservice Architecture. *IOSR Journal of Computer Engineering (IOSR-JCE)*, v. 26, n. 1, ser. 4, p. 48-58, jan.-fev. 2024. e-ISSN: 2278-0661, p-ISSN: 2278-8727.

LASZEWSKI, T.; PRAKASH, N. Migrating to the Cloud. Disponível em: <http://www.oracle.com/technetwork/articles/cloudcomp/migrating-to-the-cloud-chap-3495856.pdf>. Acesso em: 11 ago. 2014.

ORACLE CORPORATION¹. Oracle Data Pump Best Practices. *Oracle Whitepaper*, ago. 2021. Disponível em: <https://www.oracle.com/a/tech/docs/oracle-database-utilities-data-pump-bp2019.pdf>. Acesso em: 30 ago. 2024.

ORACLE CORPORATION². Oracle for SAP Database Update. *Oracle Whitepaper*, ago. 2024. Disponível em: <https://www.oracle.com/a/ocom/docs/ora4sap-dbupdate-5093030.pdf>. Acesso em: 30 ago. 2024.

SAMPATHKUMAR, J. V.; KATARI, P. Strategies for Migrating Oracle Databases to AWS. *AWS Whitepaper*, jan. 2022. Disponível em: <https://aws.amazon.com/whitepapers>. Acesso em: 25 ago. 2024.

THALHEIM, B.; WANG, Q. Data Migration: A Theoretical Perspective. *Journal of Data & Knowledge Engineering*, v. 87, p. 260-278, 2013. Disponível em: <https://doi.org/10.1016/j.datak.2012.12.003>. Acesso em: 30 ago. 2024.

VODOMIN, G.; ANDROČEC, D. Problems during Database Migration to the Cloud. *Faculty of Organization and Informatics*, Varaždin, Croácia, set. 2015.

ZHAO, J. F.; ZHOU, J. T. Strategies and methods for cloud migration. *International Journal of Automation and Computing*, v. 11, p. 143-152, mar. 2015.