MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

O USO DE CUBO DE DADOS COMO UMA SOLUÇÃO DE *BIG DATA* COMO UMA FERRAMENTA DE TOMADA DE DECISÃO

Yuri Matheus Dias Pereira

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Gerenciamento de Sistemas Espaciais. Orientada pelo Dr. Mauricio Gonçalves Vieira Ferreira e pelo Dr. Rodrigo Rocha Silva

URL do documento original:
<http://urlib.net/>

INPE
São José dos Campos
2021

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3945-6923/6921
Fax: (012) 3945-6919
E-mail: pubtc@sid.inpe.br

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

O USO DE CUBO DE DADOS COMO UMA SOLUÇÃO DE *BIG DATA* COMO UMA FERRAMENTA DE TOMADA DE DECISÃO

Yuri Matheus Dias Pereira

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Gerenciamento de Sistemas Espaciais. Orientada pelo Dr. Mauricio Gonçalves Vieira Ferreira e pelo Dr. Rodrigo Rocha Silva

URL do documento original:
<http://urlib.net/>

INPE
São José dos Campos
2021

Dados Internacionais de Catalogação na Publicação (CIP)

Sobrenome, Nomes.

Cutter O uso de Cubo de Dados como uma solução de *Big Data* como uma ferramenta de tomada de decisão / Nome Completo do Autor1; Nome Completo do Autor2. – São José dos Campos : INPE, 2021.

xxii + 49 p. ; ()

Dissertação ou Tese (Mestrado ou Doutorado em Nome do Curso) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, AAAA.

Orientador : José da Silva.

1. Palavra chave. 2. Palavra chave 3. Palavra chave. 4. Palavra chave. 5. Palavra chave I. Título.

CDU 000.000

ATENÇÃO! A FOLHA DE APROVAÇÃO SERÁ INCLUIDA POSTERIORMENTE.

Mestrado ou Doutorado em Nome do Curso

"But I try not to think with my gut. If I'm serious about understanding the world, thinking with anything besides my brain, as tempting as that might be, is likely to get me into trouble. It's OK to reserve judgment until the evidence is in.".

Carl Sagan e Ann Druyan
em *"O Mundo Assombrado pelos Demônios: A Ciência Vista Como Uma Vela no Escuro"*, 1995

AGRADECIMENTOS

- Mauricio

- Rodrigo

- Família (dos dois lados)

- Bruno e Gabriela

- Italo, Isomar e Danilo

- Comissão WETE e CubeDesign

- Jun, Pascote, Maria (?) e todos do CCS

- Seguranças do INPE (Em especial ao Eduardo)

- Membros do CITAR pelos almoços

- Todos os membros da biblioteca do INPE

Ao INPE e todos os funcionários que proveram todas a infraestrutura necessária para este trabalho.

- CAPES

- Secretaria de Pós-Graduação?

RESUMO

Satélites são monitorados pelas equipes de solo via pacotes de telemetria, que informam o estado atual dos equipamentos e permitem avaliar a capacidade do satélite de continuar a sua missão. Esses pacotes de telemetria constituem um corpo de dados de tamanho e alta complexidade, sendo que satélites que operados por vários anos geram dados históricos de grande volume, ainda úteis para as atividades de operação. O volume de dados históricos de telemetria disponíveis ao INPE atualmente é estimado em ao menos 3 *terabytes* no total, com tendência a crescer nos próximos anos. Esta proposta apresenta o uso de cubo de dados como solução para executar consultas e análises sobre esses dados. Os conceitos da área de cubo de dados são apresentados, bem como uma revisão de como outros operadores de satélite estão lidando com grandes volumes, variedades e velocidade de atualização de dados, cenário que define um contexto de *Big Data* para o domínio de controle de satélites. Devido a característica de alta dimensionalidade dos dados de telemetria, algoritmos clássicos da aŕea do cubo de dados tem dificuldade em responder consultas com resultado satisfatório para os operadores de satélite. Assim, neste trabalho é proposto identificar as consultas que são de interesse dos operadores de satélite, criar uma modelagem multidimensional para os dados de telemetria utilizando de cubo de dados, e avaliar quais são os algoritmos de construção do cubo que conseguiriam suprir as necessidades dos dados. Também são apresentados os resultados alcançados até o momento, bem como o planejamento para a continuação do trabalho.

Palavras-chave: cubo de dados. *Big Data*. Satélites. Telemetrias. Operação de Satélites.

THE USE OF A DATA CUBE AS A BIG DATA SOLUTION AS A TOOL FOR DECISION MAKING

ABSTRACT

Abstract

Keywords: Atmospheric turbulence. WETAMC campaign. LBA project. Chaotic behavior. Chaotic attractor.

LISTA DE FIGURAS

Pág.

LISTA DE TABELAS

Pág.

LISTA DE ABREVIATURAS E SIGLAS

DW	–	*Data Warehouse* (Armazém de Dados)
OLAP	–	*On-Line Analytical Processing* (Processamento Analítico Online)
OLPT	–	*On-Line Transaction Processing* (Processamento Online de Transações)
NoSQL	–	"Não apenas SQL"
TAD	–	Tipo Abstrato de Dados
ROLAP	–	*Relational OLAP*
MOLAP	–	*Multidimensional OLAP*
HOLAP	–	*Hybrid OLAP*
DBMS	–	*DataBase Management System*
TLE	–	*Two Line Element*
TID	–	*Tuple Identifier* (Identificador de Tupla)
CSV	–	Valores Separados por Vírgula
INPE	–	Instituto Nacional de Pesquisas Espaciais
CCS	–	Centro de Controle de Satélites
SCD	–	Satélite de Coleta de Dados
CBERS	–	Satélite Sino-Brasileiro de Recursos Terrestres
AMZ	–	Amazônia
NASA	–	*National Aeronautics and Space Administration*
NOAA	–	*National Oceanic and Atmospheric Administration*
L-3	–	*Level 3*
ESA	–	*European Space Operations Centre*
EUMETSAT	–	*European Organisation for the Exploitation of Meteorological Satellites*
AWS	–	*Amazon Web Services*
HDFS	–	*Hadoop Distributed FileSystem*
CSMT	–	*China Satellite Marine Track & Control Department*
SISET	–	*Shandong Institute of Space Electronic Technology*

LISTA DE SÍMBOLOS

a	–	primeira contante
b	–	segunda constante
ρ	–	densidade de um fluido
ν	–	viscosidade cinemática
R_e	–	número de Reynolds
α	–	constante de Kolmogorov
k	–	número de onda
K	–	curtose
D_0	–	dimensão de contagem de caixas
D_1	–	dimensão de informação
D_2	–	dimensão de correlação
λ_1	–	expoente de Lyapunov dominante

SUMÁRIO

Pág.

1 INTRODUÇÃO

O Centro de Controle de Satélites (CCS) é um departamento pertencente ao Instituto Nacional de Pesquisas Espaciais (INPE) atualmente monitora e controla os seguintes satélites: a família do Satélite de Coleta de Dados (SCD), composta de dois satélites SCD-1 e SCD-2, e a família do Satélite Sino-Brasileiro de Recursos Terrestres (CBERS), com apenas o quinto satélite em operação atualmente, o CBERS-4. Estes satélites realizam passagens sobre as estações terrenas do INPE, durante o qual o CCS recebe dados do estado do satélite, chamados de telemetrias, e envia telecomando, utilizados para controlar o satélite, bem como realiza atividades de manutenção e estimativa, como medidas de velocidade e posição de cada satélite (AZEVEDO; AMBRÓSIO, 2010).

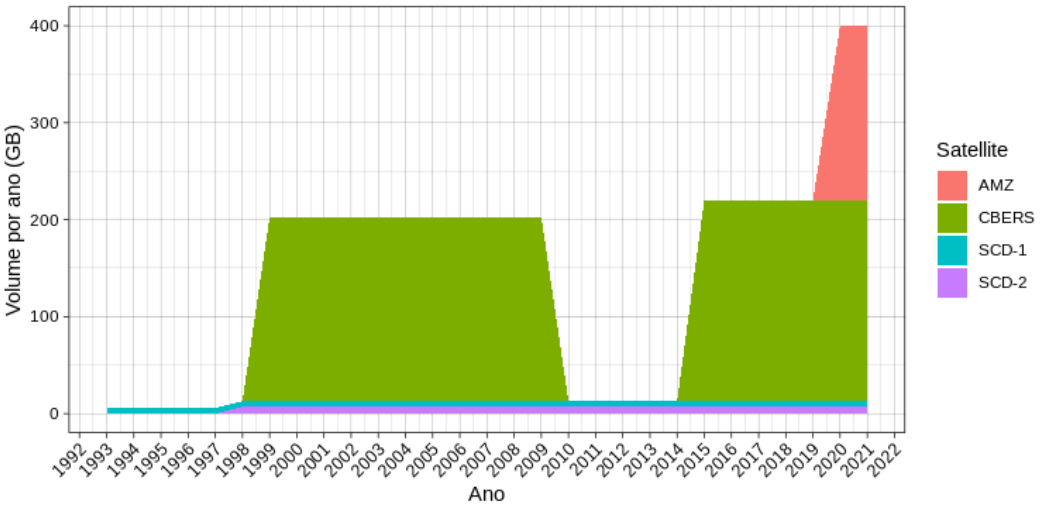
Dados de telemetria geralmente carregam medidas de sensores e verificações de saúde dos instrumentos, como temperatura das baterias, corrente de algum subsistema, se um dado equipamento está ativo ou não, bem como dados que os operadores e engenheiros acham necessários para a operação, entre outros (LARSON; WERTZ, 1999). Estes dados precisam ser guardados por toda a vida do satélite, sendo que para satélites que estão em funcionamento por vários anos adquirem um elevado volume de dados, que deve ser analisado. No caso dos satélites da família SCD, o SCD-1 já estando operacional por mais de 25 anos, e continuando a gerar dados, atualmente gera um volume aproximado de 7GB por ano.

Para satélites mais complexos como os da família CBERS, que possuem mais de 4 mil telemetrias sendo monitoradas. Com os lançamentos futuros do CBERS-4A e do Amazônia-1, o volume de dados e a complexidade da análise dos mesmos deve aumentar, criando novas necessidades de operação (FILHO et al., 2017).

A figura 1.1 mostra uma uma estimativa simples da geração histórica de dados de telemetria no CCS. Essa estimativa foi feita utilizando dos dados não compressos a partir da disponibilidade dos mesmos. Ela também assume que o Amazônia-1 vai gerar um volume de dados de telemetria similar ao gerado do CBERS.

Dessa estimativa, obtemos o total de dados de telemetria disponíveis para a análise no CCS considerando uma taxa constante dos satélites, apresentados na figura 1.2. É importante ressaltar que a grande maioria desses dados não está disponível para consulta pelo usuário, visto que somente os dados de alguns poucos anos da operação estão disponíveis para os operadores e engenheiros, necessitando de trabalho significativo para analisar dados do passado.

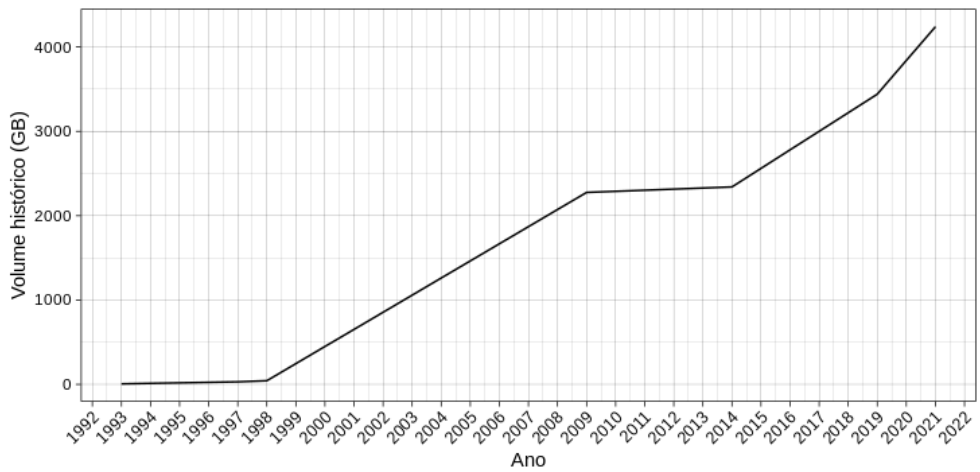
Figura 1.1 - Estimativa de geração anual de dados pelos satélites do INPE



Volume estimado de geração de dados por cada ano de operação de cada satélite.
Fonte: Produção do autor.

Esses dados devem ser propriamente tratados para que não virem "*dark data*", termo que denota quaisquer tipo de dados que não são de fácil acesso para os seus usuários em potencial (HEIDORN, 2008).

Figura 1.2 - Estimativa do volume de dados histórico de telemetria de todos os satélites



Volume total estimado de dados de telemetria gerados por todos os satélites.
Fonte: Produção do autor.

Esses dados entram na definição de *Big Data*, pois possuem um grande volume, são gerados continuamente, possuem formatos diversos, sua análise é de alto valor e existe uma incerteza quanto a qualidade dos dados devido a problemas de comunicação e degradação dos instrumentos. Essas características são denotadas pelos cinco Vs do *Big Data*: Volume, Variedade, Velocidade, Valor e Veracidade (EMANI et al., 2015).

Considerando que todos os dados já estivessem no banco de dados, propriamente formatados e prontos para a análise, ainda restariam grandes problemas: com um banco de dados na ordem dos *terabytes*, consultas sobre um número elevado de telemetrias ou que precisem de dados de vários anos poderiam demorar dias, ou mais, para serem executadas.

Deste modo, é necessário criar uma estrutura que permita a análise e consulta desses dados de uma forma estruturada e que tenha desempenho satisfatório. As tecnologias de *Data Warehouse* (DW) e *Online Analytical Processing* (OLAP) tem demonstrado capacidade e experiência para atingir esses objetivos (BIMONTE, 2016), inclusive na área espacial (YVERNES, 2018). Essas tecnologias executam a generalização de dados agregando enormes quantidade de dados em vários níveis de abstração, assim tornam elementos essenciais de apoio à decisão e atraem a atenção tanto da indústria como das comunidades de pesquisa. Sistemas OLAP, que são tipicamente dominados por consultas complexas que envolvem operadores *group-by* e operadores de agregações, são as principais características entre essas ferramentas.

Sistemas OLAP são baseados em um modelo multidimensional chamado de cubo de dados, que é uma generalização do operador *group-by* sobre todas as combinações possíveis das dimensões, com variados níveis de granularidade (GRAY et al., 1996). Cada dimensão é uma perspectiva de decisão sobre os dados, sendo formada por um subconjunto de atributos. Cada combinação é chamada de um subcubo, que correspondem a um conjunto de células descritas como tuplas sobre as dimensões do subcubo. Além das dimensões, cada tupla contém um fato, também chamado de medida, que representa o que será medido no processo de análise.

Cada dimensão pode estar organizada em uma hierarquia para facilitar a análise. Por exemplo, uma dimensão tempo pode ser dividida em "dia $<$ mês $<$ ano", com ano sendo o nível mais genérico. Essa prática visa facilitar a interpretação dos dados pelos usuários. Medidas são atributos atributos associados a uma combinação de dimensões, sendo geradas de forma estatística.

Tecnologias OLAP são caracterizadas pela habilidade em responder consultas de apoio a decisão de forma eficiente (HAN et al., 2011). Para atingir isso, o cubo de dados deve ser materializado antes da execução da consulta. Isso significa que as combinações de dimensões são computadas previamente, assim gerando o cubo de dados completo. Porém, essa abordagem possui um custo computacional exponencial em relação ao número de dimensões, assim a materialização completa do cubo envolve um grande número de células e um tempo substancial para a sua execução.

Dados de satélite são caracterizados pela sua alta dimensionalidade, onde um satélite pode precisar rastrear milhares de telemetrias. Por exemplo, supondo um satélite com $n = 100$ telemetrias, e cada telemetria representando uma dimensão, teremos 2^{100} possíveis subcubos para a implementação de um cubo de dados. Supondo uma cardinalidade, o número de valores diferentes em cada telemetria, como sendo de 100, teremos $101^{100} \approx 10^{200}$ células para cada dimensão. Devido ao controle ativo pelos operadores de satélite, os dados são concentrados em alguns valores que se repetem frequentemente, sendo que isso é chamado de *skew*.

Dessa forma, conseguir calcular e manter um cubo de dados é um problema exponencial, e reduzir o seu consumo de memória e tempo de computação é de fundamental importância para desenvolver um sistema OLAP. Para a área espacial essa necessidade é maior: a maior parte dos algoritmos de computação do cubo tem problemas em lidar com mais do que 15 dimensões (SILVA, 2015).

1.1 Objetivos

Assim, este trabalho tem como objetivo estabelecer um método para processamento de cubos de dados para a área espacial, para que o processamento de consultas OLAP sejam executadas de forma eficiente considerando-se a alta dimensionalidade, elevado número de tuplas, alto *skew* e alta cardinalidade dos dados.

Assim é necessário identificar quais são as consultas de interesse dos operadores de satélite e quais são as análises que devem ser feitas pelos mesmos. Disso será criada uma representação dimensional dos dados de telemetria em uma estrutura do cubo de dados, e algoritmos de construção do cubo devem ser avaliados para identificar qual é o mais apropriado para responder as consultas.

Como resultados esperados deste trabalho teremos a avaliação dos algoritmos de construção do cubo nos dados de alta dimensionalidade, com a adequabilidade do uso de cubo de dados como uma solução para operadores executarem as consultas

analíticas mais estratégicas para as operações de satélites.

1.2 Organização da proposta

Os capítulos restantes desta proposta estão organizados da seguinte maneira:

- Capítulo 2: Este capítulo apresenta os conceitos e fundamentos teóricos desta proposta, como os conceitos relevantes de operação de satélites, *Data Warehouse*, *Big Data* e cubo de dados.
- Capítulo 3: Neste capítulo os trabalhos correlatos de cubo de dados são apresentados, bem como as arquiteturas que outros operadores de satélite estão implementando.
- Capítulo 4: Neste capítulo a proposta é apresentada e seus conceitos principais explicados.
- Capítulo 5: Esse capítulo apresenta os resultados alcançados até o momento, apresentando os *software* utilizados.
- Capítulo 6: Com base nos resultados intermediários alcançados, esse capítulo apresentará as conclusões obtidas, bem como as direções de implementação para o resto do trabalho.

2 FUNDAMENTAÇÃO

Este capítulo apresenta os conceitos fundamentais relacionados a essa proposta, começando pela operação dos satélites na seção 2.1, apresentando a definição de *Big Data* na seção 2.2, e os os conceito de *Data Warehouse* na seção 2.3, *OLAP* na seção 2.4 e cubo de dados na seção 2.5.

2.1 Operação de satélites

Um satélite é dividido em dois módulos: o módulo de serviço e a carga útil. O módulo de serviço compõe tudo necessário para o funcionamento dos equipamentos de bordo, como o sistema de geração de energia, o sistema de comunicação com o solo, o computador de bordo, etc. A carga útil compõe todos os equipamentos necessários para cumprir os objetivos da missão, sendo esses sensores, câmeras, telescópios, etc (LARSON; WERTZ, 1999).

Um satélite gera dois tipos diferentes de dados: dados da carga útil e dados de telemetria. Os dados da carga útil são os dados gerados para cumprir a missão do satélite, sendo que eles podem ser fotos tiradas para o sensoriamento remoto, fotos tiradas por telescópios, dados de comunicação caso este seja o foco da missão entre outros (LARSON; WERTZ, 1999). Os dados de telemetria são os dados de monitoramento do estado de saúde e do funcionamento dos equipamentos do satélite. Esses dados são coletados pelo computador de bordo do satélite, e são enviados para as estações de solo via sistemas de telecomunicação.

Os dados de telemetria compõe usualmente medidas de sensores nos equipamentos do satélite, informações coletadas pelo computador de bordo (como se um instrumento está ligado ou não), e outros dados cuja coleta foi definida como relevante para a operação do satélite. Dependendo da missão, outras medidas podem ser classificadas como telemetria, como por exemplo câmeras voltadas para o satélites, radares para a detecção de possíveis colisões, etc (KRAG et al., 2017).

Esses dados devem ser analisados pelos operadores de satélite, que são os responsáveis pelo monitoramento e operação do satélite, em solo após recebimento no centro de controle. Essa análise visa garantir que o satélite está executando suas tarefas como deveria, e que o seu estado de saúde permite a continuação da missão. Neste trabalho, será utilizada a análise feita pelos operadores de satélite somente nos dados de telemetria, que é uma análise não trivial dadas as características dos dados, classificados como *Big Data*.

2.2 *Big Data*

A aplicação do conceito de *Big Data* vem evoluindo ao longo dos anos, e para este trabalho será utilizada a definição dos *5 Vs*: Volume, Variedade, Velocidade, Valor e Veracidade (EMANI et al., 2015). Em detalhes:

- **Volume**: esse termo geralmente especifica uma quantidade de dados em que um sistema tradicional de gerenciamento de banco de dados é ineficaz. É importante ressaltar que isso não se trata apenas do armazenamento dos dados, mas também do seu processamento (BOUSSOUF et al., 2018). Usar um grande volume de dados geralmente implica em modelos melhores, que então produzem análises melhores, justificando a coleta de uma grande quantidade de dados.

- **Variedade**: dados são provenientes de fontes diferentes, com formatos diferentes, sem um esquema de modelagem padronizado, como dados advindos de *logs* de computadores, dados de sensores, dados multimídia, etc. Como consequência, esses dados devem ser utilizados da forma mais transparente o possível na análise.

- **Velocidade**: dados são disponibilizados de uma forma muito rápida, e devem ser analisados da forma mais rápida o possível. Isso implica que os dados podem ser guardados e analisados até em tempo real.

- **Valor**: os dados devem ser armazenados para criar algum valor para os seus usuários, seja ele econômico, científico, social, organizacional, etc.

- **Veracidade**: os dados não possuem garantias quanto a sua qualidade, como inconsistências e falta de acurácia, porém a análise deve ser de alta qualidade de qualquer forma.

Esses V's estão relacionados com a construção de um *Data Warehouse*, sendo que também podem ser vistos como requisitos para a crição de um para um conjunto de dados caracterizado como *Big Data* (ZHANG et al., 2017). Em especial, existe um certo relacionamento com a ideia de "*NoSQL*" ("Não apenas SQL", em inglês), em que não apenas sistemas de banco de dados relacionais são utilizados, mas também outros paradigmas são utilizados, como orientados a documentos, chave e valor, etc (BIMONTE, 2016).

2.3 *Data Warehouse*

Um Armazém de Dados ou Data Warehouse (DW) é um repositório de dados orientado por assunto, integrado, variado ou particionado em função do tempo e não volátil, que auxilia no gerenciamento do processo de tomada decisões (INMON; HACKATHORN, 1994). Essa definição pode ser dividida em:

- **Orientado por assunto**: o DW é utilizado para a análise de uma área em específico. Por exemplo, é de interesse analisar especialmente os dados da carga útil de uma forma específica.
- **Integrado**: o DW deve integrar dados vindos de múltiplas fontes de uma forma estrutura. Por exemplo, mesmo que existam duas representações diferentes para um mesmo produto, o DW deve possuir apenas uma representação. Isso requer o uso de técnicas de limpeza e integração dos dados, de modo a garantir a consistência dos dados.
- **Variado em função do tempo**: o DW deve conter, explícita ou implicitamente a perspectiva de tempo. Isso quer dizer que o DW possui dados históricos e eles podem ser consultados durante a análise. Por exemplo, pode se querer saber de dados de dias, meses ou anos atrás.
- **Não volátil**: uma vez dentro do DW, os dados não são removidos ou atualizados, sendo um requisito para a consulta de dados históricos.

Essas características diferem o *Data Warehouse* de outros sistemas de repositório, como sistemas de banco de dados, sistemas de processamento de transações e sistemas de arquivos (HAN et al., 2011).

Um DW é geralmente representado por um modelo dimensional que permite eficiência na organização dos dados e na recuperação de informações gerenciais (KIMBALL; ROSS, 2013). Neste modelo são definidos fatos, dimensões e medidas. Um fato corresponde ao assunto de negócio a ser analisado, cada dimensão é uma perspectiva de visualização do assunto de negócio e medidas são valores numéricos que quantificam o assunto de negócio. Uma das dimensões é sempre temporal para permitir a análise do assunto ao longo do tempo (SILVA, 2015).

2.4 *OLAP*

On-line Analytical Processing (OLAP) é um termo que se refere a um conjunto de ferramentas que são utilizadas para resumir, consolidar, visualizar, aplicar formu-

lações e sintetizar dados de acordo com múltiplas dimensões (CODD et al., 1998).

Um sistema OLAP permite a resposta de consultas multidimensionais usando dados armazenados no *Data Warehouse* (KIMBALL; ROSS, 2013), sendo que as características principais são (BIMONTE, 2016):

- **Consultas Online**: as consultas devem ser feitas *Online*, isto é, em tempo real para o usuário.
- **Consultas Multidimensionais**: Consultas são definidas utilizando as dimensões e medidas providas pelo *Data Warehouse*, que esperam dados de alta qualidade.
- **Representação simples**: os resultados das consultas devem ser representados utilizando tabelas e gráficos, pois os usuários finais geralmente são tomadores de decisão que precisam de visualizações relevantes.
- **Exploratórias**: as consultas são utilizadas em carácter exploratório, pois geralmente os usuários não conhecem de antemão todos os dados disponíveis para consultas.

Cada ferramenta OLAP deve manipular um novo tipo abstrato de dados (TAD), chamado de cubo de dados, utilizando estratégias específicas devido ao modo de como os dados são armazenados, sendo classificadas em (MOREIRA; LIMA, 2012):

- ***Relational OLAP* (ROLAP)**: utilizam Sistemas de Gerenciamento de Banco de Dados (*Data base Management System* - DBMS) relacionais para o gerenciamento e armazenamento dos cubos de dados. Ferramentas ROLAP incluem otimizações para cada DBMS, implementação da lógica de navegação em agregações, serviços e ferramentas adicionais;
- ***Multidimensional* OLAP (MOLAP)**: implementam estruturas de dados multidimensionais para armazenar cubo de dados em memória principal ou em memória externa. Não há utilização de repositórios relacionais para armazenar dados multidimensionais e a lógica de navegação já é integrada a estrutura proposta;
- ***Hybrid* OLAP (HOLAP)**: combinam técnicas ROLAP e MOLAP, onde normalmente os dados detalhados são armazenados em base de dados relacionais (ROLAP), e as agregações são armazenadas em estruturas de dados multidimensionais (MOLAP).

É importante ressaltar a diferença entre OLAP e *Online Transaction Processing* (OLPT), visto que sistemas comuns de banco de dados utilizam apenas OLTP, que tem o objetivo de realizar transações e processar consultas online. Isso cobre a grande maioria das operações do dia a dia, como controle de estoque, operações bancárias, etc, servindo a diversos usuários de uma organização. Já o OLAP é utilizado por tomadores de decisão e analistas de dados, sendo voltado para decisões de mais alto nível na organização (HAN et al., 2011).

2.5 Cubo de dados

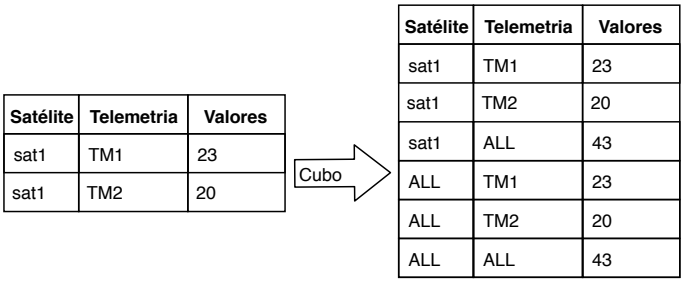
O cubo de dados originalmente foi criado como um operador relacional que gera todas as combinações possíveis de seus atributos de acordo com uma medida (GRAY et al., 1996).

A estrutura do cubo de dados permite que os dados sejam modelados e visualizados em múltiplas dimensões, e ele é caracterizado por dimensões e medidas. Uma medida é um atributo cujos valores são calculados pelo relacionamento entre as dimensões, sendo que esse é calculado utilizando funções de agragação como soma, quantidade, média, moda, mediana, etc. Uma dimensão é feita pelas entidades que compõe os nossos dados, determinando o contexto do assunto em questão (HAN et al., 2011). Uma dimensão pode ainda ser dividida em membros, que podem ter uma hierarquia, como uma divisão da dimensão tempo em dia, mês e ano.

A organização de um cubo de dados possibilita ao usuário a flexibilidade de visualização dos dados a partir de diferentes perspectivas, já que o operador gera combinações através do conceito do valor *ALL*, onde este conceito representa a agregação de todas as combinações possíveis de um conjunto de valores de atributos. Operações em cubos de dados existem a fim de materializar estas diferentes visões, permitindo busca e análise interativa dos dados armazenados (HAN et al., 2011).

Um cubo de dados é composto por células e cada célula possui valores para cada dimensão, incluindo *ALL*, e valores para as medidas. A figura 2.1 mostra um exemplo de um cubo de dados. O valor de uma medida é computado para uma determinada célula utilizando níveis de agregação inferiores para gerar os valores dos níveis de agregação superiores na estratégia *Top-down*, com a ordem inversa sendo a *Bottom-up*.

Figura 2.1 - Exemplo de um cubo de dados

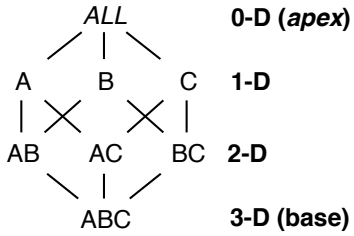


Fonte: Produção do autor.

2.5.1 Células do cubo de dados

Um cubo de dados é composto de vários subcubos, que são todos os possíveis níveis de agregação nas dimensões especificadas. Subcubos são compostos de células base e células agregadas, sendo uma célular agregada é uma célula que utiliza do valor especial ALL ("$*$") para demonstrar que está agregando valores em uma ou mais dimensões. Uma célula base não utiliza da notação ALL, sendo composta do nível mais baixo de agregação (LIMA, 2009). A figura 2.2 demonstra todos os níveis de agregação de um cubo composto das dimensões A, B e C, do mais genérico (*apex*) ao mais específico(base).

Figura 2.2 - Células de agregação em um cubo de dados



Fonte: Produção do autor.

Formalmente, supondo um cubo de dados n-dimensional, uma célula a de qualquer subcubo é definida por $a = (a_1, a_2, a_3, \ldots, a_n, medidas)$. A célula é m-dimensional (de um subcubo com m dimensões), se exatamente m, com $(m \leq n)$, valores entre $(a_1, a_2, a_3, \ldots, a_n)$ não são "$*$". Se $m = n$, então a é uma célula base, caso contrário

$(m < n)$, ela é uma célula agregada.

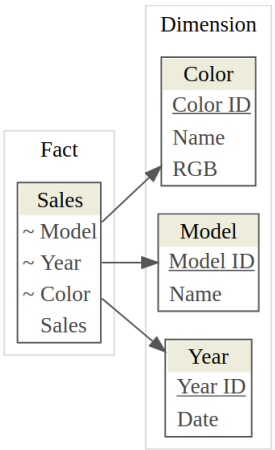
Um relacionamento de descendente-ancestral pode existir entre células. Em um cubo de dados n-dimensional, uma célula $a = (a_1, a_2, a_3, \ldots, a_n, medidas_a)$ de nível i é um ancestral de uma célula $b = (b_1, b_2, b_3, \ldots, b_n, medidbs_b)$ de nível j, e b é um descendente de a, se e somente se $i < j$ e $1 \leq m \leq n$, onde $a_m = b_m$ sempre que $a_m \neq *$. Em particular, uma célula a é chamada de pai de uma célula b, e b de filho de a, se e somente se $j = i + 1$ e b for um descendente de a (HAN et al., 2011).

2.5.2 Modelagem dimensional

Existem três esquemas principais para a modelagem dimensional de um cubo de dados: Esquema Estrela (*Star Schema*), Esquema Floco de Neve (*Snowflake Schema*) e Constelação de Fatos (*Fact Constellation Schema*).

O esquema estrela é o mais utilizado, sendo que ele contém uma tabela central chamada de tabela de fatos, onde reside a maior parte dos dados, com um conjunto menor de tabelas, chamadas de tabelas de dimensão, para as outras dimensões. A figura 2.3 mostra um exemplo de esquema estrela.

Figura 2.3 - Esquema estrela

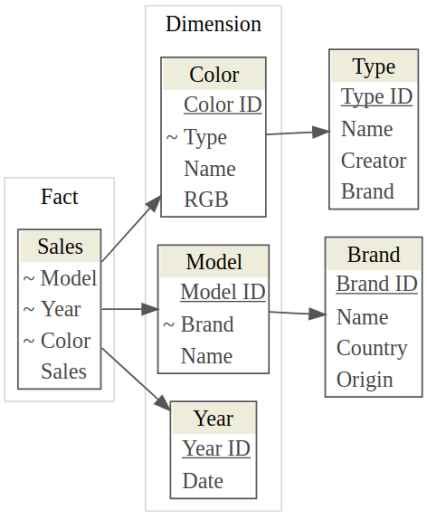


Fonte: Produção do autor.

O esquema floco de neve é uma variação do esquema estrela, onde algumas dimensões são normalizadas, dividindo os dados das tabelas de dimensão em outras tabelas. Isso possui vantagens de eliminar redundâncias nas tabelas de dimensão, porém cria

problemas durante a execução de consultas, visto que é necessário realizar operações de *join* com as novas tabelas. A figura 2.4 mostra um exemplo de esquema floco de neve.

Figura 2.4 - Esquema floco de neve



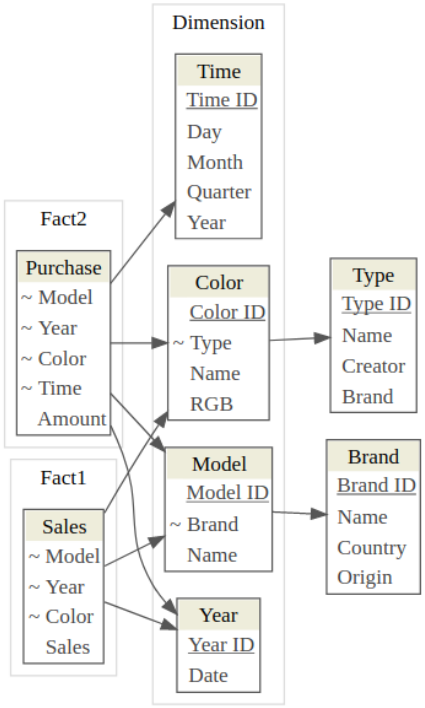
Fonte: Produção do autor.

O esquema constelação de fatos utiliza de múltiplas tabelas de fato, como se fossem várias tabelas no esquema estrela que compartilham tabelas de dimensão. Isso leva ao seu nome, como um conjunto de estrelas. A figura 2.5 mostra um exemplo de constelação de fatos.

2.5.3 Hierarquias de conceito

Uma hierarquia de conceitos é utilizada para definir uma sequência de mapeamento entre um conjunto de conceitos de baixo nível para um conjunto de conceitos de alto nível, mais gerais. É um estilo de agrupamento e discretização, pois agrupa os valores de modo a reduzir a cardinalidade de uma dimensão (HAN et al., 2011). Elas ajudam a tornar a análise mais fácil de ser entendida, pois as operações traduzem os dados de baixo nível em uma representação que é mais fácil para o usuário final, assim facilitando a execução das consultas e o seu subsequente uso.

Figura 2.5 - Esquema constelação de fatos



Fonte: Produção do autor.

2.5.4 Medidas

Cada célula de um cubo é definida como um par $\langle (d_1, d_2, \ldots, d_n), medidas \rangle$, onde $(d_1, d_2, \ldots, d_n)$ representam as combinações possíveis de valores de atributos sobre as dimensões. Uma medida é calculada para uma certa célula agregando os dados correspondentes a combinação de dimensões e valores (HAN et al., 2011). Medidas podem ser classificadas em três tipos: distributiva, algébrica e holística.

Uma medida distributiva é uma medida cujo cálculo pode ser particionado e depois combinado, e o resultado seria o mesmo se o cálculo fosse executado em todo o conjunto de dados. Por exemplo, a função de soma é distributiva: dividindo os dados N em conjuntos n, e fazendo a soma de cada conjunto n, teremos o mesmo resultado que se a fosse feita diretamente sobre N.

Uma medida algébrica é uma medida cujo cálculo pode ser feito sobre duas ou mais medidas distributivas. Por exemplo, uma medida de média pode ser calculada com a divisão da medida *soma* pela a medida *contagem*, que são ambas distributivas.

Uma medida é holística se não existe uma medida algébrica com M argumentos que caracterize a computação. Isso quer dizer que a computação não pode ser particionada, com valores exatos obtidos apenas se a medida for aplicada em todos os dados. Alguns exemplos são as medidas de moda, desvio padrão e mediana (HAN et al., 2011).

2.5.5 Operações OLAP

Para realizar consultas no *Data Warehouse*, é necessário utilizar de algumas operações sobre o cubo de dados para obter os resultados adequados. Essas consultas também devem conseguir passar na hierarquia de conceitos de cada dimensão, bem como seguir o modelo dimensional do cubo definido, para conseguir oferecer uma interface amigável com o usuário para análise interativa (HAN et al., 2011). Algumas operações estão exemplificados na figura 2.6, porém elas geralmente são:

- *Roll-up*: realiza agregação no cubo de dados, seja navegando na hierarquia de conceitos de nível específico para um mais genérico, ou reduzindo uma dimensão.

- *Drill-down*: o inverso da operação *roll-up*, navega na hierarquia de conceitos do nível mais genérico para o nível mais específico, ou adiciona dimensões ao cubo atual. Essa operação visa aumentar o nível de detalhes dos dados.

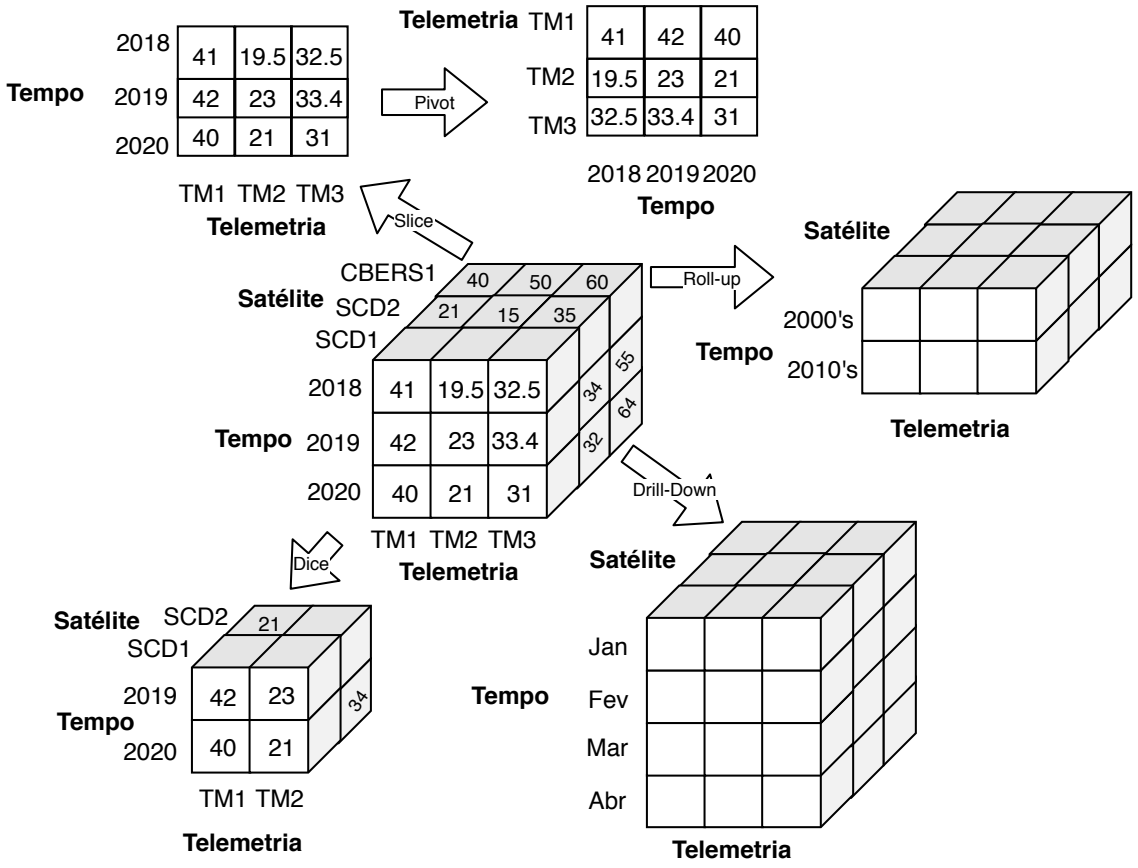
- *Slice*: ou "fatiamento", realiza uma seleção em uma dimensão do cubo, resultando em um subcubo.

- *Dice*: define um subcubo realizando uma seleção (*slice*) em duas ou mais dimensões.

- *Pivot*: também chamada de rotação, permite mudar a posição das dimensões na visualização, portanto alterando linhas por colunas e vice-versa.

Dependendo do sistema OLAP, é possível que outras operações sejam possíveis, como *drill-across* que passa por mais do que uma tabela de fatos, e *drill-through* que permite executar consultas direto na representação em baixo nível do cubo (HAN et al., 2011).

Figura 2.6 - Operações *OLAP* em um cubo de dados



Fonte: Produção do autor.

2.5.6 Computação do cubo de dados

A computação do cubo de dados é uma tarefa essencial, pois a pré-computação de todo ou parte de um cubo de dados pode aumentar significativamente o desempenho do DW. Porém, essa tarefa possui complexidade exponencial em relação ao número de dimensões, sendo chamada de materialização, com a materialização completa exigindo uma grande quantidade de células, e portanto um elevado consumo de memória e tempo (HAN et al., 2011).

O cálculo original da computação do cubo de dados foi proposta por (GRAY et al., 1996), sendo: dada uma relação de relação de entrada R com tuplas de tamanho n, o número de subcubos que podem ser gerados é 2^n, onde n é o número de dimensões do cubo. Por exemplo, supondo um cubo com três dimensões *Satélite, Telemetria, Valor*, teremos $2^3 = 8$ subcu-

bos possíveis: $\{(satélite, telemetria, valor), (satélite, valor), (satélite, telemetria), (telemetria, valor), (telemetria), (valor), (satélite), ()\}$, com () denotando o agrupamento vazio (célula base, as dimensões não estão agrupadas).

Porém, na prática, as dimensões podem possuir hierarquias de conceito associadas, como para a dimensão tempo: "dia<mês<trimestre<semestre<ano". Para um cubo com n dimensões com múltiplas hierarquias de conceito, o número total de subcubos é apresentado na equação 2.1.

$$subcubos = \prod_{i=1}^{n}(L_i + 1) \tag{2.1}$$

Onde L_i é o número de níveis de conceito da dimensão i. É necessário adicionar um a equação 2.1 para denotar o nível virtual ALL. O tamanho de cada subcubo também depende da cardinalidade de cada dimensão, isto é, o número de valores distintos. Enquanto o número de dimensões, hierarquias de conceito e cardinalidade do cubo aumenta, também aumentam os seus requisitos de espaço de forma exponencial, sendo conhecida como a **maldição de dimensionalidade** na computação do cubo (HAN et al., 2011).

Para conseguir responder as consultas de maneira apropriada, é necessário escolher um método para a computação dos subcubos: a não materialização, a materialização completa e a materialização parcial.

Na não materialização, os subcubos agregados não são pré-computados, assim as agregações são computadas imediatamente, que podem ser extremamente lentas, porém tem o menor consumo de memória.

A materialização completa computa todos as agregações possíveis do cubo, gerando um cubo de dados completo. Esse método gera os melhores tempos de resposta, pois as agregações já foram computadas, porém necessita de uma grande quantidade de espaço de memória.

A materialização parcial computa apenas um subconjunto selecionado de subcubos, sendo que existem diversas técnicas diferentes de seleção dos subcubos que serão computados. Uma delas é computar todos os subcubos que contém apenas células que satisfazem um dado critério, especificado pelo usuário. Esses cubos são chamados de *iceberg* (BEYER; RAMAKRISHNAN, 1999).

Outra técnica é computar cubos pequenos, geralmente entre 3 e 5 dimensões, para formar cubos completos. Para responder consultas com mais dimensões, as combinações entre os subcubos pequenos são agregadas. Esta técnica é chamada de *shell fragment*, e o cubo é chamado de *cube shell* (LI et al., 2004).

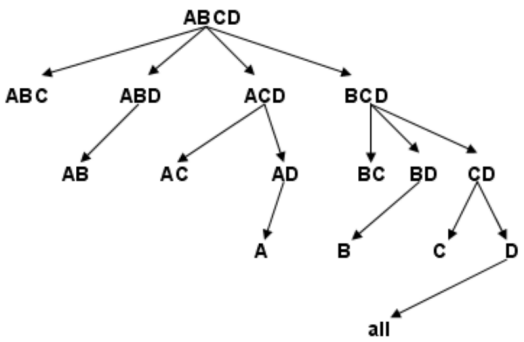
Um cubo de dados onde as células com medidas idênticas são encapsuladas em uma única abstração, chamada de célular fechada (*closed cell*) é chamado de cubo fechado, ou cubo quociente. Esta técnica foi apresentada com o cubo fechado (*closed cube*) (Dong Xin et al., 2006) e com o cubo quociente (*quotient cube*) (LAKSHMANAN et al., 2002).

A escolha da materialização parcial depende do equilíbrio necessário entre tempo de resposta e espaço de armazenamento. Porém, a computação do cubo completo continua sendo relevante, sendo que os avanços na computação dos cubos parciais são geralmente adotados na computação do cubo completo. Existe ainda o problema de atualização do cubo, pois cada atualização pode causar uma recomputação parcial ou completa do cubo para manter as medidas corretas.

A partir de um cubo base, a computação do cubo de dados pode utilizar a estratégia *Top-down* ou *Bottom-up* para a geração dos subcubos remanescentes (HAN et al., 2011).

A figura 2.7 mostra a geração de um cubo de dados de quatro dimensões pela estratégia *Top-down*. Sendo ABCD um cubo base, os subcubos de três dimensões são: ABC, ABD, ACD e BCD; que podem utilizar os resultados do cubo base para serem computados.

Figura 2.7 - Computação de cubo de dados através da estratégia *Top-Down*

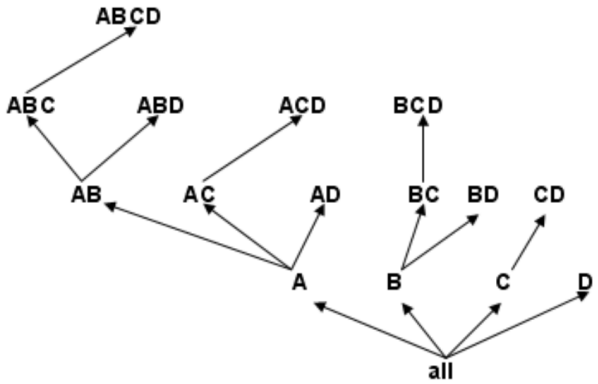


Fonte: (SILVA, 2015).

Os resultados da computação do subcubo ACD podem ser utilizados para computar AD, que consequentemente podem ser utilizados para computar A. Essa computação compartilhada permite que a estratégia *Top-down* compute agregações em múltiplas dimensões. Os valores agregados intermediários podem ser reutilizados para a computação de subcubos descendentes sucessivos.

A Figura 2.8 mostra a geração de um cubo de dados de 4 dimensões por meio da estratégia *Bottom-up*. Subcubos de poucas dimensões tornam-se pais de subcubos com mais dimensões. Infelizmente, a computação compartilhada, utilizada na estratégia *Top-down*, não pode ser aplicada quando utilizada a estratégia *Bottom-up*, então cada subcubo descendente necessita ser computado do início.

Figura 2.8 - Computação de cubo de dados através da estratégia *Bottom-up*



Fonte: (SILVA, 2015).

3 TRABALHOS CORRELATOS

Nessa seção são apresentados os trabalhos correlatos a essa proposta, que podem ser divididos em duas seções: as soluções de *Big Data* de outros operadores, e os algoritmos existentes de construção do cubo de dados com alta dimensionalidade.

3.1 Dados da operação

A tabela 3.1 mostra os tipos de dados relevantes para a operação, a sua origem e o seu formato esperado. Essa tabela considera apenas dados considerados como telemetria do próprio satélite, ou dados advindos de terceiros.

Tabela 3.1 - Dados de Operação

Tipo de Dado	**Origem**	**Formato**
Sensores de bordo	Equipamentos no satélite	Tabelas, CSV
Registros do Computador	Computador de Bordo	Texto (*Logs*)
Multimídia	Câmeras	MP4, JPG, RAW
Parâmetros orbitais	Operação, Rastreio	TLE, texto, tabelas
Documentação associada	Operadores, engenharia	Texto (Word, Excel)
Clima Espacial	Sensores no solo ou espaço	Texto, tabelas, avisos
Situational Awareness	Radares, US-STRACOM, etc	Texto, tabelas, avisos

Fonte: Adaptado de (ZHANG et al., 2017)

Para este trabalho, apenas os dados vindos de sensores de bordo serão considerados. Os outros dados nesta tabela poderiam ser considerados para uma *Data Warehouse* mais completa, porém estão fora do escopo desta proposta.

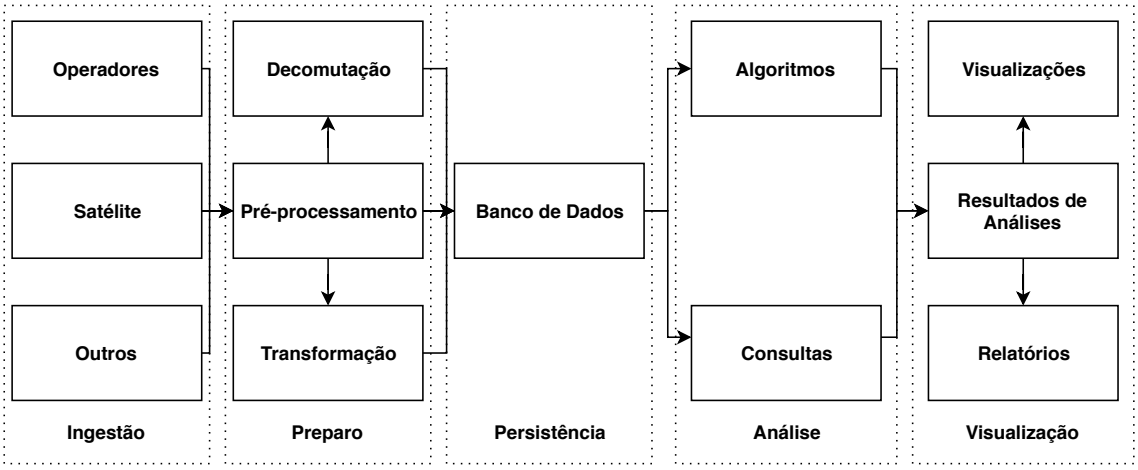
3.1.1 Fluxo dos dados

Baseado nos trabalhos correlatos e nos dados levantados, a figura 3.1 demonstra o fluxo de dados esperado de uma arquitetura de *Big Data* para a operação de satélites.

Este fluxo está separado em cinco etapas que vão desde a origem dos dados até o seu resultado de análise, e este trabalho visa apenas mapear qual seria esse fluxo baseado nos trabalhos correlatos. Cada uma das etapas está detalhada a seguir:

- **Ingestão**: onde os dados serão coletados na sua fonte (satélites, sensores no solo, outras fontes, etc). Essa etapa trata de **onde** estão os dados e **como**

Figura 3.1 - Fluxo de dados em uma arquitetura de *Big Data*



Fonte: Adaptado de (ZHANG et al., 2017)

.

coletá-los, bem como **quais** são os dados importantes de serem coletados. A "fonte" aqui pode ser um serviço de terceiros, dentro da própria instituição ou disponível de outra forma.

- **Preparo**: os dados relevantes são selecionados, e transformações são realizadas para inserir os mesmos na base de dados. Essa etapa trata do formato específico dos dados, realizando operações de limpeza, verificação da qualidade e da relevância para a análise, entre outras. O seu objetivo é garantir que os dados tem qualidade, relevância, e estão no formato adequado para a base de dados.

- **Persistência**: após o devido processamento, os dados de alta qualidade são guardados em uma base de dados, de onde ficarão disponíveis para a análise. Nessa etapa um banco de dados é utilizado, tratando apenas em como esses dados estão guardados e como eles serão disponibilizados para as consultas e execução de algoritmos.

- **Análise**: nesta etapa são executadas as consultas e os algoritmos de interesse para a análise. Podem ser desde consultas simples ("qual era o valor da telemetria X durante a passagem Y?"), a execução de algoritmos complexos ("preveja os valores da telemetria X para a próxima passagem").

- **Visualização**: os resultados das consultas e algoritmos são visualizados. Podem conter desde gráficos simples, como um histograma de uma telemetria, a relatórios complexos de um subsistema/satélite, bem como resultados de algoritmos.

Os trabalhos de (ZHANG et al., 2017), (MATEIK et al., 2017) e (BOUSSOUF et al., 2018) definem esse processo mais claramente dentre os trabalhos apresentados.

3.2 Análise de dados em outros operadores de satélite

A tabela 3.2 mostra uma revisão feita em artigos recentes sobre os operadores de satélite e quais tecnologias eles estão utilizando para atingir objetivos semelhantes, principalmente com o uso de *Big Data*.

Os objetivos em comum desses trabalhos são facilitar as atividades dos operadores por meio de algoritmos de detecção de anomalias e de verificação dos limites nos valores das telemetrias. Alguns dos operadores dessa lista estão responsáveis pela operação de constelações de satélites complexos, como constelações de sensoriamento remoto, que faz necessário um certo nível de automação ou a operação contínua teria um custo inviável.

Nesses trabalhos, o uso dessas tecnologias é apenas para os operadores de satélite, pois em nenhum desses trabalhos eles estão na mesma estrutura de ingestão dos dados da carga útil, mesmo utilizando as mesmas tecnologias, como demonstrado em (MATEIK et al., 2017) e (ADAMSKI, 2016).

Alguns desses trabalhos não utilizam de estruturas completas que seguem um fluxo de dados, como é o caso de (FERNÁNDEZ et al., 2017) e (TROLLOPE et al., 2018) que utilizam de *scripts* feitos de forma *ad-hoc*, não mostrando uma visão da arquitetura completa do fluxo de dados e apenas a ferramenta utilizada para análise pontual.

O trabalho de (YVERNES, 2018) utiliza de estratégias OLAP e do cubo de dados, tendo utilizado uma modelagem dimensional para a operação de uma constelação de satélites, porém esse trabalho menciona apenas em alto nível a modelagem utilizada, e menciona que o trabalho foi somente na parte da modelagem dimensional e integração dos dados utilizando ferramentas já existentes.

Tabela 3.2 - Operadores de Satélite e Arquiteturas de Big Data

Referência	Operador	Ferramenta	Tecnologias
(ADAMSKI, 2016)	L3 (EUA)	InControl	Hadoop, Spark, HBase, MongoDB, Cassandra, Amazon AWS
(BOUSSOUF et al., 2018)	Airbus	Dynaworks	Hadoop, Spark, HDFS, HBase, PARQUET, HIVE
(SCHULSTER et al., 2018)	EUMETSAT	CHART	MATLAB, MySQL, Oracle
(ZHANG et al., 2017)	SISET (China)	-	Hadoop, HDFS, PostgreSQL, MongoDB, Logstash, Kibana, ElasticSearch, Kafka, MapReduce
(YVERNES, 2018)	Telespazio France	PDGS	OLAP (DataCube), Saiku, Pentaho, Jaspersoft OLAP
(DISCHNER et al., 2016)	SwRI + NOAA	CYGNSS MOC	SFTP, -
(EDWARDS, 2018)	EUMETSAT	MASIF	FTP, RESTful service, JMS Messague Queue, PostgreSQL
(EVANS et al., 2016)	S.A.T.E + ESA/ESOC	-	Java, CSV
(FEN et al., 2016)	CSMT (China)	-	não menciona as tecnologias
(TROLLOPE et al., 2018)	EUMETSAT	CHART	algoritmos ad-hoc, estudo de caso
(GILLES, 2016)	L-3	InControl	Amazon EC2, LXC, Nagios
(HENNION, 2018)	Thales Alenia	AGYR	Logstash, Kafka, InfluxDB, ElasticSearch, Kibana, Grafana
(MATEIK et al., 2017)	Stinger, NASA	-	Logstash, ElasticSearch, Kibana, HDF5, CSV, R, Python, AWS, Excel
(FERNÁNDEZ et al., 2017)	NASA	MARTE	R, CSV, ad-hoc

Fonte: Produção do autor.

3.2.1 Análise de dados no INPE

O INPE já realiza análise de dados em outros departamentos, inclusive sobre as telemetrias de satélite. Os operadores devem monitorar os valores das telemetrias e informar a engenharia caso apareça algum problema que não pôde ser corrigido (TOMINAGA et al., 2017). Um exemplo está no trabalho (MAGALHÃES, 2012), feito sobre uma falha no satélite CBERS-2, onde o modelo proposto visa melhorar o conhecimento sobre avalanche térmica nas baterias para impedir que isso aconteça novamente em outros satélites. A motivação principal dos trabalhos da tabela 3.2 era a detecção de anomalias, que teve alguns algoritmos estudados em (AZEVEDO et al., 2011).

Outros setores, utilizam a análise de dados vindos da carga útil do satélite ou de agentes externos ao INPE, como dados de sensoriamento remoto, cuja análise não é trivial e também estão classificados como *Big Data*. Monteiro (2017) utilizam de conceitos de Big Data para análise de trajetórias de objetos; Ramos et al. (2016) demonstram o uso de softwares como o Hadoop para a análise de dados do clima espacial, com uma arquitetura relacionada as arquiteturas revisadas na seção anterior; e Simões et al. (2018) mostra uma arquitetura que utiliza de cubo de dados para a análise de séries temporais no sensoriamento remoto.

3.3 Computação do cubo de dados

A computação seletiva do cubo de dados possui muitos algoritmos diferentes implementados, porém eles possuem dificuldades no trato de dados com muitas dimensões e no uso limitado da memória (HAN et al., 2011).

O *FragCubing* (LI et al., 2004) apresenta o conceito de *cube shells*, onde subcubos com poucas dimensões (de 3 a 5 neste exemplo) são calculados utilizando de índices invertidos, que funcionam apenas utilizando memória principal. A ideia principal é decompor o cubo original em fragmentos que podem ser reunidos eficientemente para responder uma consulta multidimensional.

Precursor para o computação distribuída do cubo, (DOKA et al., 2011) apresenta o *Brown Dwarf*, um sistema *Peer-to-Peer* que permite atualização das células, desenhado para diminuir a redundância em cubos distribuídos.

O *PopUp-Cubing* é apresentado em (HEINE; ROHDE, 2017), que utiliza de icebergs para lidar com dados em formato de *stream*, obtendo resultados superiores ao FTL e *Star-Cubing*. Este trabalho é de interesse especial por utilizar de dados de *stream*,

que permitiriam resultados parecidos com tempo real, que são mais parecidos com os dados disponíveis para a operação de satélites, porém este cenário não será abordado neste trabalho.

Com foco em *Big Data* e utilizando como base o esquema *MapReduce*, (WANG et al., 2013) apresenta o algoritmo *HaCube* para computação do cubo em paralelo. Este trabalho apresenta um balanço entre computação do cubo em paralelo por vários nós de *MapReduce*, que permite algumas atualizações e computação incremental de medidas. Devido a própria natureza distribuída, ele precisa de mecanismos de tolerância a falha, e também os testes foram executados com no máximo apenas 5 dimensões, porém com até 2,4 bilhões de tuplas. Ainda na linha do *MapReduce*, (YANG; HAN, 2017) demonstra a computação de medidas holísticas apresentando o *Multi-RegionCube*, porém realizando menos testes que o *HaCube*.

Em (ZHAO et al., 2018) é apresentado o *Closed Frag-Shells Cubing*, que utiliza de uma combinação da abordagem de cubos fechados com a abordagem *Shell fragments*, obtendo resultados melhores que a aplicação de cada uma delas separadamente. Essa abordagem utiliza de índices *bitmap* e índices invertidos, sendo que lidam com dados altamente dimensionais e sem uma hierarquia de forma similar ao necessário neste trabalho.

qCube (SILVA et al., 2013) estende a abordagem *FragCubing* para permitir consultas sobre intervalos de valor, estendendo os operadores de consultas clássicas em cubo de dados além do operador de igualdade.

HFrag (SILVA et al., 2015) apresenta o uso de memória externa na computação dos índices invertidos, utilizando de um sistema híbrido de memória para armazenar as partições do cubo tanto na memória principal quanto na secundária, com os valores mais frequentes sendo armazenados na memória principal e os valores menos frequentes na memória secundária.

A abordagem *Hybrid Inverted Cubing* (HIC) (SILVA et al., 2016) estende a abordagem *HFrag* com o parâmetro de frequência acumulada crítica, obtendo resultados melhores do que este nas mesmas consultas.

Destes trabalhos, o *FragCubing* continua sendo um algoritmo robusto para a computação do cubo, com suas técnicas de índice invertido sendo utilizadas e ainda obtendo resultados adequados. Porém, Li et al. (2004) ilustram o impacto exponencial no consumo de memória nas diferentes abordagens de computação de cubos de

dados usando apenas 12 dimensões, sendo que há uma saturação quando cubos com 20, 50 ou 100 dimensões são computados utilizando abordagens de cubos completos, cubos DWARF, MCG, cubos fechados ou quocientes (SILVA, 2015).

3.3.1 *FragCubing*

O *FragCubing* (LI et al., 2004) apresenta o conceito de inversão de tupla. Cada tupla invertida iT tem um valor de atributo, uma lista de identificadores da tupla (TIDs) e um conjunto de valores de medida. Por exemplo, consideremos quatro tuplas: $t_1 = (tid_1, a_1, b_2, c_2, m_1)$, $t_2 = (tid_2, a_1, b_3, c_3, m_2)$, $t_3 = (tid_3, a_1, b_4, c_4, m_3)$, e $t4 = (tid_4, a_1, b_4, c_1, m_4)$. Estas quatro tuplas geram oito tuplas invertidas: $iTa_1, iTb_2, iTb_3, iTb_4, iTc_1, iTc_2, iTc_3$ e iTc_4, demonstradas na figura 3.2.

Para cada valor de atributo é construído uma lista de ocorrências, assim para a_1 temos $iTa_1 = (a_1, tid_1, tid_2, tid_3, tid_4, m_1, m_2, m_3, m_4)$ onde o valor de atributo a 1 está associado aos TIDs: tid_1, tid_2, tid_3, e tid_4. O identificador de tupla tid_1 tem o valor de medida m_1, tid_2 tem o valor de medida m_2, tid_3 tem o valor de medida m_3, e tid_4 possui o valor de medida m_4. A consulta $q = (a_1, b_4, COUNT)$ pode ser respondida por $iTa_1 \cap iTb_4 = (a_1b_4, tid_3, tid_4, COUNT(m_3, m_4))$. Em $q, iTa_1 \cap iTb_4$ indica os TIDs comuns em iTa_1 e iTb_4.

Figura 3.2 - Exemplo de uma tabela dimensional e a respectiva lista de índices invertidos

TID	A	B	C	m
tid1	a1	b2	c2	m1
tid2	a1	b3	c3	m2
tid3	a1	b4	c4	m3
tid4	a1	b4	c1	m4

Valor	Lista de TIDs	Medidas
a1	tid1, tid2, tid3, tid4	m1, m2, m3, m4
b2	tid1	m1
b3	tid2	m2
b4	tid3, tid4	m3, m4
c1	tid4	m4
c2	tid1	m1
c3	tid2	m2
c4	tid3	m3

Fonte: Produção do autor.

A complexidade da interseção é proporcional ao número de ocorrências de um valor de atributo, mais precisamente é igual ao tamanho da menor lista. Neste exemplo, iTb_2 com um TID é a menor lista. O número de TIDs associado a cada valor de atributo pode ser enorme, assim relações com dimensões de baixa cardinalidade e elevado número de tuplas necessitam de alta capacidade de processamento. Listas de TIDs pequenas permitem que consultas sejam respondidas rapidamente, portanto relações com baixo *skew* e alta cardinalidade são mais adequadas de serem computadas pela abordagem *FragCubing*.

Skew pode ser definido como o grau de uniformidade dos valores de atributos numa relação, sendo que *skew* zero indica relação com valores de atributos uniformemente distribuídos, e quanto maior o *skew* menos uniformemente distribuída a relação se encontra.

4 Experiment

- The data available, how the next chapters are related and the brief description of the data come here. - Also describe how the experiments will be performed and what will be measured. Basically the section of the results from the fragpaper.

5 Query Partition

- The method and reasoning behind trying to do this?

5.1 Algorithm and heuristic description

- Describe the algorithm for calculating the multi dimensional distance between time series

- Vanishing gradient problem?

- Lacking proper review of this part of the work, not sure if this chapter should be this complete or not

- Need to be careful with algorithm details

5.2 Queries

- Describe each query, how they are related and basically section 2 of the Information paper

5.3 Experimental Verification

- Basically overview the results of the information paper, but much more quickly and succintly. - Not even sure if can use the full text of it, or the graphics, but will have to do.

6 IntervalFrag

This section describes the IntervalFrag algorithm, and the proposed architecture needed to implement the enhancements to the FragCubing's algorithm.

6.1 Using Intervals in Inverted Indexes

- What problem are we trying to solve?

- The idea

6.2 Algorithm

- Simple implementation overview

- Insertion in the index

- Using iceberg conditions

- The Intersection problem and algorithm

- The skew influence

- Mention that there are ways to improve the algorithm further, and that they are further mentioned in appendix A

6.3 Results

- Have this section here or coalesce on the next one? Might be too big already

7 Analysis and Discussion

- The famous table of when to use this algorithm or not that Rodrigo so much wants

8 CONCLUSIONS

This work shows that it is possible to further optimize data cube algorithms by gathering information from the underlying data, and how this can be made to aid the end user's experience by decreasing implementation requirements and improving response times.

8.1 Main contributions

One of the stated purposes of this work was to find ways of using the data's domain characteristics to improve the satellite operator's day to day activities, and this work has achieved three main results:

a) A heuristic to discover related telemetries between satellite time series data and how to use this with the help of an operator to validate the relevant queries;

b) Using the previous heuristic to enhance FragCubing's query response time and memory by pre-partitioning the data;

c) Improving upon FragCubing's Inverted Index memory model by saving only intervals instead of the entire values, and thus reducing memory and query response times for some queries;

8.2 Future work

The natural evolution of this work would be to test it using other data cube algorithms, as there's a great variety of them mentioned in section 3 and not all of them might be applicable to satellite telemetry data, or showcase useful performance metrics. On that note the use of bCubing (SILVA, 2015) will be interesting, as the inverted index separation into blocks can further improve upon the memory usage as described in this chapter.

The use of the gathered satellite data on other projects is also of interest, as there's no public reliable dataset of satellite telemetry data that contains all relevant data and not just a subset of a subsystem, and this work showcases a volume that has information enough for the training of Machine Learning and Artificial Intelligence projects. Only projects that release full telemetry data are relatively simple CubeSat projects, who do not generate a significant volume that is enough for the use of these

algorithms. The author plans to release the dataset in a citable format for the use of the community in the near future.

This work also has the potential of improving query execution when dealing with multiple satellites, constellations and/or formations, it needing only the data to be gathered and the suitable cube format defined to be tested.

Furthermore the Set Intersection problem defined in chapter 6 can be further optimized with recent advances not only in computer architectures, but also with regards to complexity and the validation of the algorithms in real world datasets. A preliminary investigation was performed, as a simple but not rigorous overview of the results is detailed in Annex .

8.3 Final thoughts

The approach and architecture detailed in this work...

REFERÊNCIAS BIBLIOGRÁFICAS

ADAMSKI, G. Data Analytics for Large Constellations. In: **SpaceOps 2016 Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2016. (SpaceOps Conferences). 23, 24

AZEVEDO, D. N. R.; AMBRÓSIO, A. M. Dependability in Satellite Systems: An Architecture for Satellite Telemetry Analysis. In: WORKSHOP EM ENGENHARIA E TECNOLOGIA ESPACIAIS, 1. (WETE)., 30 mar. - 1 abr. 2010, São José dos Campos. **Anais...** São José dos Campos: INPE, 2010. IWETE2010-1065, p. 6. ISSN 2177-3114. 1

AZEVEDO, D. N. R.; AMBRÓSIO, A. M.; VIEIRA, M. **Estudo sobre técnicas de detecção automática de anomalias em satélites**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011. 25

BEYER, K.; RAMAKRISHNAN, R. Bottom-up Computation of Sparse and Iceberg CUBE. In: **Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data**. Philadelphia, Pennsylvania, USA: ACM, 1999. (SIGMOD '99), p. 359–370. ISBN 978-1-58113-084-3. 18

BIMONTE, S. Open issues in Big Data Warehouse design. **Revue des Nouvelles Technologies de l'Information**, p. 10, 2016. 3, 8, 10

BOUSSOUF, L.; BERGELIN, B.; SCUDELER, D.; GRAYDON, H.; STAMMINGER, J.; ROSNET, P.; TAILLEFER, E.; BARREYRE, C. Big Data Based Operations for Space Systems. In: **2018 SpaceOps Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2018. 8, 23, 24

CODD, E. F.; CODD, S.; SALLEY, C. Providing olap to user-analysts: An it mandate. In: . [S.l.: s.n.], 1998. 10

DISCHNER, Z.; REDFERN, J.; ROSE, D.; ROSE, R.; RUF, C.; VINCENT, M. CYGNSS MOC; Meeting the challenge of constellation operations in a cost-constrained world. In: **2016 IEEE Aerospace Conference**. [S.l.: s.n.], 2016. p. 1–8. 24

DOKA, K.; TSOUMAKOS, D.; KOZIRIS, N. Brown Dwarf: A fully-distributed, fault-tolerant data warehousing system. **Journal of Parallel and Distributed Computing**, v. 71, n. 11, p. 1434–1446, nov. 2011. ISSN 0743-7315. 25

Dong Xin; Zheng Shao; Jiawei Han; Hongyan Liu. C-Cubing: Efficient Computation of Closed Cubes by Aggregation-Based Checking. In: **22nd International Conference on Data Engineering (ICDE'06)**. [S.l.: s.n.], 2006. p. 4–4. 19

EDWARDS, T. Dealing with the Big Data - The Challenges for Modern Mission Monitoring and Reporting. In: **15th International Conference on Space Operations**. Marseille, France: American Institute of Aeronautics and Astronautics, 2018. ISBN 978-1-62410-562-3. 24

EMANI, C. K.; CULLOT, N.; NICOLLE, C. Understandable Big Data: A survey. **Computer Science Review**, v. 17, p. 70–81, ago. 2015. ISSN 1574-0137. 3, 8

EVANS, D. J.; MARTINEZ, J.; Korte-Stapff, M.; VANDENBUSSCHE, B.; ROYER, P.; RIDDER, J. D. Data Mining to Drastically Improve Spacecraft Telemetry Checking: A Scientist?s Approach. In: **SpaceOps 2016 Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2016, (SpaceOps Conferences). 24

FEN, Z.; YANQIN, Z.; CHONG, C.; LING, S. Management and Operation of Communication Equipment Based on Big Data. In: **2016 International Conference on Robots Intelligent System (ICRIS)**. [S.l.: s.n.], 2016. p. 246–248. 24

FERNÁNDEZ, M. M.; YUE, Y.; WEBER, R. Telemetry Anomaly Detection System Using Machine Learning to Streamline Mission Operations. In: **2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)**. [S.l.: s.n.], 2017. p. 70–75. 23, 24

FILHO, A. C. J.; AMBRÓSIO, A. M.; FERREIRA, M. G. V.; LOUREIRO, G. The Amazonia-1 satellite's ground segment - challenges for implementation of the space link extension protocol services. In: INTERNATIONAL ASTRONOMICAL CONGRESS, 68. (IAC), 25-29 Sept., Adelaide, Australia. **Proceedings...** [S.l.], 2017. p. 1–12. 1

GILLES, K. Flying Large Constellations Using Automation and Big Data. In: **SpaceOps 2016 Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2016, (SpaceOps Conferences). 24

GRAY, J.; BOSWORTH, A.; LYAMAN, A.; PIRAHESH, H. Data cube: A relational aggregation operator generalizing GROUP-BY, CROSS-TAB, and

SUB-TOTALS. In: . [S.l.]: IEEE Comput. Soc. Press, 1996. p. 152–159. ISBN 978-0-8186-7240-8. 3, 11, 17

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques, Third Edition**. 3 edition. ed. Haryana, India; Burlington, MA: Morgan Kaufmann, 2011. ISBN 978-93-80931-91-3. 4, 9, 11, 13, 14, 15, 16, 17, 18, 19, 25

HEIDORN, P. B. Shedding Light on the Dark Data in the Long Tail of Science. **Library Trends**, v. 57, n. 2, p. 280–299, 2008. ISSN 1559-0682. 2

HEINE, F.; ROHDE, M. PopUp-Cubing: An Algorithm to Efficiently Use Iceberg Cubes in Data Streams. In: **Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies**. Austin, Texas, USA: ACM, 2017. (BDCAT '17), p. 11–20. ISBN 978-1-4503-5549-0. 25

HENNION, N. Big-data for satellite yearly reports generation. In: **2018 SpaceOps Conference**. [S.l.]: American Institute of Aeronautics and Astronautics, 2018. 24

INMON, W. H.; HACKATHORN, R. D. **Using the Data Warehouse**. Somerset, NJ, USA: Wiley-QED Publishing, 1994. ISBN 978-0-471-05966-0. 9

KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling**. Edição: 3rd. Indianapolis, IN: John Wiley & Sons, 2013. ISBN 978-1-118-53080-1. 9, 10

KRAG, H.; SERRANO, M.; BRAUN, V.; KUCHYNKA, P.; CATANIA, M.; SIMINSKI, J.; SCHIMMEROHN, M.; MARC, X.; KUIJPER, D.; SHURMER, I.; O'CONNELL, A.; OTTEN, M.; MUÑOZ, I.; MORALES, J.; WERMUTH, M.; MCKISSOCK, D. A 1 cm space debris impact onto the Sentinel-1A solar array. **Acta Astronautica**, v. 137, p. 434–443, ago. 2017. ISSN 0094-5765. 7

LAKSHMANAN, L. V. S.; PEI, J.; HAN, J. Quotient Cube: How to Summarize the Semantics of a Data Cube. In: **Proceedings of the 28th International Conference on Very Large Data Bases**. Hong Kong, China: VLDB Endowment, 2002. (VLDB '02), p. 778–789. 19

LARSON, W. J.; WERTZ, J. R. (Ed.). **Space Mission Analysis and Design, 3rd Edition**. 3rd edition. ed. El Segundo, Calif. : Dordrecht ; Boston: Microcosm, 1999. ISBN 978-1-881883-10-4. 1, 7

LI, X.; HAN, J.; GONZALEZ, H. High-dimensional OLAP: A Minimal Cubing Approach. In: **Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30**. Toronto, Canada: VLDB Endowment, 2004. (VLDB '04), p. 528–539. ISBN 978-0-12-088469-8. 19, 25, 26, 27

LIMA, J. d. C. **SEQUENTIAL AND PARALLEL APPROACHES TO REDUCE THE DATA CUBE SIZE**. Tese (Doutorado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 2009. 12

MAGALHÃES, R. O. de. **Estudo de avalanche térmica em um sistema de carga e descarga de bateria em satélites artificiais**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, São José dos Campos, fev. 2012. 25

MATEIK, D.; MITAL, R.; BUONAIUTO, N. L.; LOUIE, M.; KIEF, C.; AARESTAD, J. Using Big Data Technologies for Satellite Data Analytics. In: . [S.l.]: American Institute of Aeronautics and Astronautics, 2017. ISBN 978-1-62410-483-1. 23, 24

MONTEIRO, D. V. **A FRAMEWORK FOR TRAJECTORY DATA MINING**. Tese (Doutorado), 2017. 25

MOREIRA, A. A.; LIMA, J. d. C. Full and partial data cube computation and representation over commodity PCs. In: **2012 IEEE 13th International Conference on Information Reuse Integration (IRI)**. [S.l.: s.n.], 2012. p. 672–679. 10

RAMOS, M. P.; TASINAFFO, P. M.; de Almeida, E. S.; ACHITE, L. M.; da Cunha, A. M.; DIAS, L. A. V. Distributed Systems Performance for Big Data. In: LATIFI, S. (Ed.). **Information Technology: New Generations**. [S.l.]: Springer International Publishing, 2016, (Advances in Intelligent Systems and Computing). p. 733–744. ISBN 978-3-319-32467-8. 25

SCHULSTER, J.; EVILL, R.; PHILLIPS, S.; FELDMANN, N.; ROGISSART, J.; DYER, R.; ARGEMANDY, A. CHARTing the Future – An offline data analysis and reporting toolkit to support automated decision-making in flight-operations. In: **15th International Conference on Space Operations**. Marseille, France: American Institute of Aeronautics and Astronautics, 2018. ISBN 978-1-62410-562-3. 24

SILVA, R. R. **Abordagens para Cubo de Dados Massivos com Alta Dimensionalidade Baseadas em Memória Principal e Memória Externa:**

HIC e BCubing. Tese (Doutorado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 2015. 4, 9, 19, 20, 27, 37

SILVA, R. R.; HIRATA, C. M.; LIMA, J. d. C. A Hybrid Memory Data Cube Approach for High Dimension Relations. In: HAMMOUDI, S.; MACIASZEK, L. A.; TENIENTE, E. (Ed.). **ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015**. [S.l.]: SciTePress, 2015. p. 139–149. ISBN 978-989-758-096-3. 26

_____. Computing BIG data cubes with hybrid memory. **Journal of Convergence Information Technology**, v. 11, n. 1, p. 18, jan. 2016. 26

SILVA, R. R.; LIMA, J. d. C.; HIRATA, C. M. qCube: Efficient integration of range query operators over a high dimension data cube. **JIDM**, v. 4, n. 3, p. 469–482, 2013. 26

SIMÕES, R. E. d. O.; CAMARA, G.; QUEIROZ, G. R. de. Sits: Data analysis and machine learning using satellite image time series. In: Workshop de Computação Aplicada, 18. (WORCAP), 21-23 ago., São José dos Campos, SP. **Resumos...** [S.l.], 2018. p. 18. 25

TOMINAGA, J.; FERREIRA, M. G. V.; AMBRÓSIO, A. M. Comparing satellite telemetry against simulation parameters in a simulator model reconfiguration tool. In: CERQUEIRA, C. S.; BÜRGER, E. E.; YASSUDA, I. d. S.; RODRIGUES, I. P.; LIMA, J. S. d. S.; OLIVEIRA, M. E. R. de; TENÓRIO, P. I. G. (Ed.). **Anais...** São José dos Campos: Instituto Nacional de Pesquisas Espaciais (INPE), 2017. ISSN 2177-3114. 25

TROLLOPE, E.; DYER, R.; FRANCISCO, T.; MILLER, J.; GRISO, M. P.; ARGEMANDY, A. Analysis of automated techniques for routine monitoring and contingency detection of in-flight LEO operations at EUMETSAT. In: **2018 SpaceOps Conference**. Marseille, France: American Institute of Aeronautics and Astronautics, 2018. ISBN 978-1-62410-562-3. 23, 24

WANG, Z.; CHU, Y.; TAN, K.-L.; AGRAWAL, D.; ABBADI, A. E.; XU, X. Scalable Data Cube Analysis over Big Data. **arXiv:1311.5663 [cs]**, nov. 2013. 26

YANG, H.; HAN, C. Holistic and Algebraic Data Cube Computation Using MapReduce. In: **2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. [S.l.: s.n.], 2017. v. 2, p. 47–50. 26

YVERNES, A. Copernicus Ground Segment as a Service: From Data Monitoring to Performance Analysis. In: **15th International Conference on Space Operations**. Marseille, France: American Institute of Aeronautics and Astronautics, 2018. ISBN 978-1-62410-562-3. 3, 23, 24

ZHANG, X.; WU, P.; TAN, C. A big data framework for spacecraft prognostics and health monitoring. In: **2017 Prognostics and System Health Management Conference (PHM-Harbin)**. [S.l.: s.n.], 2017. p. 1–7. 8, 21, 22, 23, 24

ZHAO, Q.; ZHU, Y.; WAN, D.; TANG, S. A Closed Frag-Shells Cubing Algorithm on High Dimensional and Non-Hierarchical Data Sets. In: **Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication**. Langkawi, Malaysia: ACM, 2018. (IMCOM '18), p. 6:1–6:8. ISBN 978-1-4503-6385-3. 26

APPENDIX A - INTERSECTION ALGORITHMS

This is only a simple overview to show that the problem is important and has a lot of research behind

- Overview of the intersection algorithms used

- The results from the intersection algorithms

- What to test now with other algorithms (Ding, Li, etc with references!)

- Summary of the algorithms: Table A.1. All wrong for now

Tabela A.1 - Set Intersection Algorithms

Algorithm	**Reference**	**Complexity**	
Scalar	Cormen?	$O(n+M)$	
Li	?	$O(n+M)$	
std::set_intersect	C++ impl?	$O(n+M)$	
HashSet	Not even cormen can save me	$O(n+M)$	
UnorderedSet	-	$O(n+M)$	
BinaryIntersection	-	$O(n+M)$	
BinaryIterator	-	$O(n+M)$	
SIMD (SS2)	-	$O(n+M)$	
Ding?	-	$O(n+M)$	

ANEX A - PUBLICATIONS

This annex showcases the publications that resulted from this work, and from the general Master's effort. Table A.1 shows the summary of the published, and currently expecting to be published articles.

Tabela A.1 - Resulting published work

Name	QUALIS	SCOPUS Percentile	Source	Status
WETE 2018	NA - Conference	NA		Published
IAC 2019	NA - Conference	NA		Published
MDPI Information	B2 (A4)	47%	-	Accepted, later retracted
IEEE Latin America Transactions	B2	61%	-	Submitted
IntervalFrag	-	-	-	Writing
Inverted Index Intersection in Data Cubes	-	-	-	Writing

Furthermore, the table shows that the last two articles are being written, and will be published shortly with the results derived from this work.

Tem os artigos do CubeDesign + Jenny/LIT que faltam aqui, coloco eles ou não? Eu tive pouca atuação, não sei se é relevante colocar aqui

PUBLICAÇÕES TÉCNICO-CIENTÍFICAS EDITADAS PELO INPE

Teses e Dissertações (TDI)

Teses e Dissertações apresentadas nos Cursos de Pós-Graduação do INPE.

Notas Técnico-Científicas (NTC)

Incluem resultados preliminares de pesquisa, descrição de equipamentos, descrição e ou documentação de programas de computador, descrição de sistemas e experimentos, apresentação de testes, dados, atlas, e documentação de projetos de engenharia.

Propostas e Relatórios de Projetos (PRP)

São propostas de projetos técnico-científicos e relatórios de acompanhamento de projetos, atividades e convênios.

Publicações Seriadas

São os seriados técnico-científicos: boletins, periódicos, anuários e anais de eventos (simpósios e congressos). Constam destas publicações o Internacional Standard Serial Number (ISSN), que é um código único e definitivo para identificação de títulos de seriados.

Pré-publicações (PRE)

Todos os artigos publicados em periódicos, anais e como capítulos de livros.

Manuais Técnicos (MAN)

São publicações de caráter técnico que incluem normas, procedimentos, instruções e orientações.

Relatórios de Pesquisa (RPQ)

Reportam resultados ou progressos de pesquisas tanto de natureza técnica quanto científica, cujo nível seja compatível com o de uma publicação em periódico nacional ou internacional.

Publicações Didáticas (PUD)

Incluem apostilas, notas de aula e manuais didáticos.

Programas de Computador (PDC)

São a seqüência de instruções ou códigos, expressos em uma linguagem de programação compilada ou interpretada, a ser executada por um computador para alcançar um determinado objetivo. Aceitam-se tanto programas fonte quanto os executáveis.