UFRRJ
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

DISSERTAÇÃO

Tratamento e preenchimento de falhas de séries de dados meteorológicos utilizando workflows científicos paralelos em ambientes de GPU

Fábio Cardozo da Silva

2014

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXTAS
CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

TRATAMENTO E PREENCHIMENTO DE FALHAS DE SÉRIES DE DADOS METEOROLÓGICOS UTILIZANDO WORKFLOWS CIENTÍFICOS PARALELOS EM AMBIENTES DE GPU

FÁBIO CARDOZO DA SILVA

Sob a Orientação do Professor
Sérgio Manuel Serra da Cruz

e Co-orientação da Professora
Priscila Machado Lima Vieira

Dissertação submetida com requisito parcial para a obtenção do grau de **Mestre em Ciências**, no Curso de Pós-Graduação em Modelagem Matemática e Computacional

Seropédica, RJ
Setembro 2014

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÀTICA E COMPUTACIONAL

FÁBIO CARDOZO DA SILVA

Dissertação submetida como requisito parcial para obtenção do grau **de Mestre em Ciências**, no Curso de Pós-Graduação em Modelagem Matemática e Computacional, área de Concentração em Ciências Exatas.

DISSERTAÇÃO APROVADA EM 18/09/2014

_____

Sérgio Manuel Serra da Cruz. (Dr. em Engenharia de Sistemas de Computação. UFRRJ

_____

Ronaldo Ribeiro Goldschmidt. Dr. em Engenharia Elétrica. IME

_____

Gustavo Bastos Lyra. Dr.em Agronomia. UFRRJ

Agradecimentos

Sem dúvida alguma, o primeiro agradecimento deve ser feito ao Senhor da criação, Deus, que é o grande orquestrador de todas as dádivas. Ao Senhor, muito obrigado.

Em seguida quero agradecer a minha mãe (Sandra Cardozo da Silva), e a minha avó (Marli Casanova da Silva), que são as responsáveis pela minha primeira educação. Sendo a minha avó a pessoa que me apresentou, com todo o carinho, a forma de se vencer as condições mais adversas.

As palavras a seguir, são dedicadas a minha musa de todos os dias, a pessoa que optou por dividir o seu espaço, a sua vida e momentos comigo. Que me confiou a honrosa missão de ser, seu esposo, amigo e chato particular. Laura Paula Cardozo, muito obrigado, por seu carinho e compreensão.

Agora é a vez de agradecer a elas, as minhas meninas, Ana Júlia Paula Cardozo e Ana Vitória Paula Cardozo. Com certeza, vocês não perceberam, mas ajudaram muito ao pai, na conclusão desse trabalho. Em meio a brincadeiras, risos e choros construímos juntos essa história.

RESUMO

SILVA, Fábio Cardozo da. **Tratamento e preenchimento de falhas de séries de dados meteorológicos utilizando workflows científicos paralelos em ambientes de GPU. 2014**. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Programa de Pós-Graduação em Modelagem Matemática e Computacional, Departamento de Matemática, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2014.

Juntamente com a crescente importância das pesquisas na área de meteorologia e climatologia, principalmente as que manipulam grandes volumes de dados voltados aos estudos dos recursos hídricos, surgem as dificuldades para que os pesquisadores dessas áreas obtenham e armazenem dados de alta qualidade em seus repositórios. Este trabalho tem como objetivo apresentar uma proposta na área computacional capaz de processar dados meteorológicos agregando controle de qualidade a longas séries históricas de dados em hidrologia. Os artefatos deste trabalho são baseados na visão da e-Science, utilizando *workflows* científicos em ambientes de processamento de alto desempenho que tem por finalidade automatizar parte das etapas de pesquisas científicas em meteorologia. Além disso, este trabalho propõe a integração de *workflows* científicos desenvolvidos na plataforma VisTrails com a computação paralela em ambientes GPU utilizando códigos CUDA. Essa integração visa ampliar a capacidade de manipulação de grandes volumes de dados hidrológicos. Outra característica desse trabalho são a apresentação dos ganhos de desempenho da solução computacional e a representação dos dados relativos à proveniência retrospectiva dos experimentos segundo os moldes da especificação PROV-DM. Como um dos principais resultados temos o índice de identificação e correção de falhas de 87,7%, nos testes realizados com 77 estações, o que representa um ganho precioso de tempo na preparação de dados nas pesquisas da área. Com isso pode-se concluir que a combinação da visão da e-Ciência associada a tecnologia de computação paralela CUDA, além de viável, se torna uma alternativa no tratamento de grandes volumes de dados na área de Meteorologia e Climatologia.

Palavras-chaves: Meteorologia, Climatologia, *e-Ciência*, Proveniência, *Workflows* Científicos, Hidrologia, Computação paralela.

ABSTRACT

SILVA, Fábio Cardozo da. **Tratamento e preenchimento de falhas de séries de dados meteorológicos utilizando workflows científicos paralelos em ambientes de GPU. 2014**. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Programa de Pós-Graduação em Modelagem Matemática e Computacional, Departamento de Matemática, Universidade Federal Rural do Rio de Janeiro, Seropédica, 2014.

Despite of the growing importance of researches in the field of Meteorology, especially those that handle large volumes of data focused on studies of hidrological resources, difficulties the handle datasets are increasing. Researchers are developing great efforts to obtain and store high quality data in their repositories. This dissertation aims to present a computational proposal capable to compute meteorological data and add quality control to long times series of data. The artifacts conceived and developed in this work are based on the e-Science vision. We used high performance processing features and scientific workflows to aid to automate the process of scientific research in Meteorology. Furthermore, this work integrates VisTrails scientific workflows with parallel computing environments using GPU and CUDA programming. The integration was guided to extend the capability of handling large volumes of hihg quality meteorological data. Other features of this work are the discussions about performance gains of the proposal and the representation of (raw and curated) data and retrospective provenance generated by the computational experiments according to PROV-DM specification. The main results of this work are. 87,7% of detection of errors and failures replacemente were achieved using 77 meteorological stations. We can conclude that the fusion of E-Sceince vision with CUDA parallel computing approach is viable to deal with large volumes of meterological and climatological data.

Keywords: Meteorology, Climatology, *e-Science*, Provenance, Scientific Workflows, Hidrology Paralel Computing.

LISTA DE FIGURAS

ÍNDICE DE TABELAS

SUMÁRIO

INTRODUÇÃO

A Meteorologia e Climatologia, são ciências bastante aplicadas que atendem diversas atividades humanas que necessitam das informações sobre o tempo e o clima, como por exemplo: defesa civil; agricultura; aviação e navegação (civil e militar); setor energético e de gerenciamento de recursos hídricos; estudos de impacto ambiental, controle de poluentes; planejamento comercial e econômico; setor de seguros; atividades de turismo, esporte e de espetáculos, dentre outras. Além disto, as pesquisas científicas realizadas em Meteorologia e Climatologia são de grande importância para a sociedade tanto no que diz respeito à prevenção dos desastres naturais de origem atmosférica quanto no planejamento ambiental e sócio-econômico a curto, médio e longo prazo (FENG; HU; QIAN, 2004).

Atualmente as pesquisas em Meteorologia e Climatologia necessitam se desenvolver em ambientes computacionais que ofereçam velocidades cada vez maiores, e um dos principais responsáveis por esse desenvolvimento são diversas formas de computação distribuídas e paralelas. A obtenção de uma diversidade de dados meteorológicos através de sensores ou a geração de dados simulados por modelos meteorológicos e climáticos, propicia a geração de repositórios de dados de formatos e semânticas distintas que armazenam grandes volumes de dados que por si só, servem como um novo objeto de pesquisa. O fenômeno da criação e crescimento veloz de grandes volumes de dados a partir da medidas de sensores e, ou consequente necessidade de manipulação de dados se repete e amplia a cada dia. Hoje conhecemos esse fenômeno por Big Data (HEY; TOLLE, 2009).

Uma das possíveis formas de se conduzir pesquisas em e-Science, focada em Meteorologia e Climatologia, que manipule grandes volumes de dados é através da utilização de *workflows* Científicos. Segundo Cruz (CRUZ, 2011), os *workflows* científicos estão se disseminando em diversos tipos de projetos e-Science e passam a ser amplamente utilizados em pesquisas científicas e em projetos interdisciplinares, cujo o objetivo de automatizar etapas de um experimento, o que permite a composição de diversos programas a fim de ordenar essas etapas e se alcançar o objetivo de pesquisa (DEELMAN et al., 2009).

Com isso podemos considerar que a Meteorologia é uma das Ciências que mais se beneficiam do consórcio entre os temas computação de alto desempenho, proveniência e *workflows* científicos, além de possuirem grande influência e importância nos contexto social, econômico e agroambiental (FENG; HU; QIAN, 2004).

No entanto, em pesquisas na Meteorologia e Climatologia, existem grandes problemas associados com obtenção e a qualidade de dados. O número de repositórios abertos que disponibilizam dados meteorológicos é vasto, e estão sujeitos a diversos tipos de falhas que variam desde questões operacionais (disponibilidade do dado) até questões mais intrinsecamente relacionados ao item de dado, tais como falhas nas coletas das séries históricas. A detecção e correção de falhas exigem grandes esforços envolvendo o uso de diversas técnicas de pré-processamento de dados, preparação de dados e disponibilização de dados para que as pesquisas possam ser efetivamente realizadas.

Com isso, pode-se caracterizar que o problema de pesquisa a ser abordado nesta dissertação, é o estudo e desenvolvimento de métodos computacionais para o preenchimento de falhas e controle de qualidade de dados, nas series históricas, existentes nos arquivos de dados pluviométricos do banco de dados do sistema HIDROWEB da Agência Nacional de

Águas (ANA), mais especificamente nos dados de 89 estações do estado do Rio de Janeiro. Para que tanto o preenchimento de falhas, quanto o controle de qualidade, realizados pela execução de *workflows* científicos, conta com a coleta de dados de proveniência, combinados a técnicas de computação paralela em ambientes de processadores GPU (*Graphic Process Unit*).

1.2 Objetivo Geral

O objetivo geral desse trabalho é estudar e consequente construir uma solução computaçional, que melhore a qualidade dos dados meteorológicos e seja capaz de realizar o pré-processamento e o preenchimento de falhas nas séries históricas de dados meteorológicos coletados em estações meteorológicas, mais especificamente pluviométricas, localizadas nas nas seis mesorregiões do estado do Rio de Janeiro, que se utilize de *workflows* científicos e aplicando técnicas de computação, baseado em programação paralela em GPU. A partir dessas ferramentas será criado um banco de dados, compostos de informações meteorológicas curadas da região de estudo, essas informações serão confiáveis e de qualidade.

1.3 Objetivos Específicos

Os objetivos específicos desse trabalho são:

- Desenvolver *workflows* científicos, utilizando a plataforma Vistrails (SGWFC), para o pré-processamento de dados meteorológicos a partir dos estudos prévios de Filho et al (2013).
- Utilizar banco de dados, que armazenará os dados brutos de cada estação. Esses dados devem ser submetidos às técnicas de correção das falhas, através de métodos estatísticos definidos por Filho et al (2013).
- Utilizar de mecanismo de coleta dos metadados de proveniência, tanto da qualidade, quanto da correção de falhas, como das etapas de desenvolvimento do *workflow*.
- Implementar de solução computacional que se utilize de processamento paralelo em arquitetura CUDA para acelerar o processamento dos dados meteorológicos.
- Estudar e análisar desempenho dos *workflows* em ambiente de GPU, além de estudos comparativos no que diz respeito às formas de execução dos workflows (execução sequencial *vs*. paralela).
- Analisar os experimentos e resultados, quanto à acurácia dos dados processados pelos *workflows* (avaliação da correção de falhas nas séries históricas).

2 FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA

Nesse capítulo serão apresentados os principais temas e conceitos envolvidos nesta dissertação, e as comparações com principais trabalhos relacionados.

2.1 Meteorologia

A meteorologia é uma Ciência que têm grande influência e importância nos contexto social, econômico e agroambiental (INMET, 2013), Como tempo, clima e do ciclo hidrológico não conhecem fronteiras geopolíticas, a cooperação internacional à escala global é essencial para o desenvolvimento da meteorologia e hidrologia operacional, assim como para pleno aproveitamento dos benefícios de sua aplicação.

A meteorologia gera conhecimentos importantes para diversos aspectos da vida humana. No Brasil, trabalhos importantes fornecem informações, para o desenvolvimento e implantação de políticas públicas na área de saúde, como no caso de estudos sobre os índices de raios ultra violeta (DE PAULA CORRÊA, 2005) e sua publicação em meios de fácil acesso, pois a exposição a raios ultra violeta são a causa de inúmeros casos de câncer. Estudos como estes tornam se relevantes, para o governo e entidades relacionadas.

Outra área importante em constante estudo por meteorologistas, é a hidrologia, por conta da importância do ciclo das águas, estudos geram informações consideráveis sobre as chuvas, seu comportamento, ou da sua ausência (seca), essas informações servem de norte, para áreas como agricultura (DA SILVA et al., 2011), ou mesmo para área de defesa civil e planejamento urbano, se observadas as áreas regiões que comumente sofrem com enchentes (ALVES FILHO; RIBEIRO, 2006).

O principal organismo internacional que trata de questões ligadas à Meteorologia é a Organização Meteorológica Mundial (OMM) (*World Meteorological Organization* WMO em inglês). A OMM é uma agência especializada das Nações Unidas que lidera os estudos sobre estado e comportamento da atmosfera da Terra, sua interação com os oceanos, o clima que produz e a distribuição resultante dos recursos hídricos. A OMM tem uma adesão de 191 Estados-Membros e Territórios, tornando-se a agência especializada das Nações Unidas. Para a meteorologia (tempo e clima), hidrologia operacional e ciências geofísicas relacionadas. em 1951, com sua fundação em 1950.

No Brasil, as organizações que tratam da meteorologia, se dividem entre os níveis Federal e Estadual, sendo o principal o Instituto Nacional de Meteorologia – INMET, órgão ligado diretamente ao Ministério da Agricultura, Pecuária e Abastecimento (MAPA). O INMET visa o fornecimento de informações meteorológicas confiáveis, com o objetivo de influir construtivamente no processo de tomada de decisões do país (“INMET”, 2013).

No estado do Rio de Janeiro, estado alvo desse trabalho, tem-se o Sistema de Meteorologia Estatual – SIMERJ, criado em 1996. O SIMERJ têm como objetivo o monitoramento do tempo e climatologia do estado do Rio de Janeiro (SIMERJ, 2014). No entanto os dados utilizados nesses estudos, são da Agência Nacional de Águas – ANA, que possui uma extensa base de dados de cobertura nacional, os dados disponilibilizados pela ANA, são as observações de estação meteorológicas, onde são coletadas informações de chuvas.

Nesse trabalho serão utilizados os dados de precipitação pluvial, disponibilizados pela ANA, através do sistema HidroWeb. Os dados envolvidos nesse estudo são as séries históricas das estações pluviométricas, sob responsabilidade da ANA (existem outros órgãos que sedem dados para o HidroWeb).

Pesquisas em meteorologia e climatologia, devem atender o uso de intervalo mínimo de dados climatológicos, a fim de estabelecer a compatibilidade entre os dados coletados, esse padrão é conhecido como *Normal Climatológica*, e compreende o intervalo de 30 anos. Sendo estabelecido em 1872 pelo Comitê Meteorológico Internacional (INMET, 2013).

Tomando como base os conceitos anteriores, pode-se dizer que meteorologia, assim como outras áreas das ciências exatas, existe a necessidade de se utilizarem ou desenvolverem novas técnicas de computação que sejam capazes de tratar de modo adequado os grandes volumes de dados meteorológicos que se caracterizam pela elevada heterogeneidade, diferentes formatos e presença de falhas. A partir das informações até aqui apresentadas é possível entender com clareza que pesquisas em Meteorologia geram e consomem grandes volumes de dados em cada um dos seus experimentos, principalmente quando mediados pelo computador, utilizando dados coletados de forma semi e ou automática (no caso das estações meteorológicas) e ao encerrar uma pesquisa, é possível que se tenha um volume final de dados maior que o inicial. Diversas sub-áreas da Meteorologia, como por exemplo a Hidrologia, possuem necessidades que vão além da manipulação e do grande volume de dados e da necessidade do uso de técnicas de processamento intensivo de dados e de proveniência de dados. Com isso pode-se dizer que dados climatológicos são importantes para o entendimento de mudanças e acontecimentos em uma região, além de serem fundamentais na tomada de decisões nas áreas agrícola e hidrológica (FENG; HU; QIAN, 2004).

2.1.1 Hidrologia

A Hidrologia é definida, como a ciência que estuda a água na Terra, como ocorre, forma de circulação e distribuição, propriedades físicas e químicas e a relação com o meio ambiente (EAGLESON, 1993).

Nesta dissertação serão utilizadas as séries de dados hidrológicos, obtidos a partir da leitura pluviométrica, que é a medição da quantidade de água em estado líquido que é transferida da atmosfera ao solo, fenômeno conhecido como precipitação, das estações meteorológicas distribuídas pelo estado Rio de Janeiro, essas séries possuem os dados de observação diária e a média mensal das chuvas. A medição é expressa em milímetros (mm), e é conhecida como altura da chuva, essa altura pluviométrica (h), é definida pelo volume precipitado em uma unidade de área horizontal de determinado terreno, sendo:

$$h = \frac{1\,\text{litro de água}}{1\,\text{m}^2\,\text{de terreno}} = \frac{1000\,\text{cm}^3}{10000\,\text{cm}^2} = 0{,}1\,\text{cm} = 1\,\text{mm de chuva}$$

Figura 1- Cálculo da altura pluviométrica.

Esses dados são de relevância para estudos hidrológicos, portanto, são essas séries que devem ser investigadas e tratadas para produzirem dados de qualidades, No entanto, estes não ocorrem com frequência nos repositórios de dados atualmente disponíveis nas séries históricas.

2.1.2 Séries Históricas de dados meteorológicos

Os dados utilizados nesta pesquisa fazem parte de séries de dados meteorológicos coletados em diversas estações distribuídas na mesoregião sul fluminense do estado do Rio de Janeiro. Na sua maioria, as estações pluviométricas estão sob responsabilidade da Fundação Superintendência Estadual de Rios e Lagoas (SERLA), do Instituto Nacional de Meteorologia (INMET), do Serviço Geológico do Brasil (CPRM) e da LIGHT (Light Centrais Elétricas).

Os dados das séries climáticas de precipitação pluvial foram obtidos a partir de arquivos de textos disponíveis no banco de dados da Agência Nacional de Águas - ANA, com auxílio da ferramenta HIDROWEB (http://hidroweb.ana.gov.br). A estrutura de um arquivo de dados é apresenta na Figura 2.

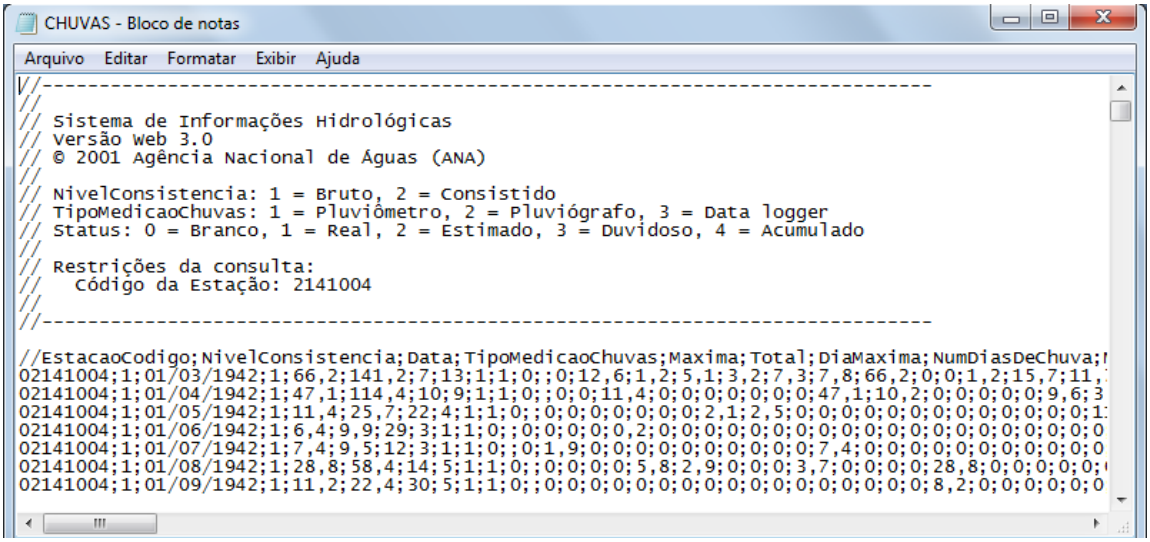

Figura 2 – Estrutura dos dados sistema HidroWeb

Os principais dados disponibilizados pelo sistema HidroWeb (figura 2) dizem respeito a: estações meteorológicas (nome, código, município, latitude, longitude, altitude, intervalo de tempo das observações, tipo de estação, responsável e operadora). Os atributos para cada mês de observações de cada estação são: nível de consistência, data de medição, tipo de medição, máxima, total, dia da máxima, número de dias de chuvas além de um atributo “chuva” para cada dia do mês.

As séries utilizadas são públicas e são as mesmas utilizadas o artigo de (FILHO et al., 2013), eles dizem respeito às estações localizadas entre as seguintes coordenadas, latitudes 22º 03’ e 23º 21’ S e longitudes 43º 25’ e 44º 54’ W, e abrangendo toda a região Sul Fluminense do estado do Rio de Janeiro, Brasil.

2.1.3 Qualidade de Dados

Em diversas áreas de estudos, a qualidade dos dados é de suma importância, ainda mais quando se trabalham com séries históricas de dados baseados em sistemas de sensoriamento coletados de forma automatizada ou manual baseados na observação humana. Essas características somadas há existência de extensas séries, envolvendo diversas fontes, como

inúmeras estações meteorológicas, diversos órgãos e institutos gestores que possuem dados similares, com métricas e formatações diferenciadas. O que pode dificultar imensamente as pesquisas e influenciar negativamente seus resultados. Além de erros contidos nessas séries históricas podem ocorrer por falhas de equipamento ou falha humana, podendo ocorrer também ambos os casos (FILHO et al., 2013; LIMA; SANTOS, 2004).

Na meteorologia, o controle da qualidade podem ser realizados por meio de filtros capazes de identificar erros na coleta de dados realizada por meio de sensores (FENG; HU; QIAN, 2004). Esses filtros devem ser capazes de identificar diversos tipos de falhas, por exemplo, mínimos e máximo espúrios, sem excluir extremos que sejam válidos na série de dados, realizando análise entre eventos temporalmente isolados, além da análise e comparação de dados isoladas baseado em tempo e espaço (MAGINA, 2007).

As avaliações de qualidade de dados meteorológicos pode ser iniciada com a eliminação de valores extremos, considerados absurdos para determinada região. Com isso impedem que esses dados sejam usados em qualquer operação relativa a região em investigação. No entanto, essa é uma forma pouco refinada para promover a qualidade de dados. Uma outra forma de promover a qualidade de dados é a localização de uma estação próxima à região com dados considerados extremos, e partir dos dados dessa segunda região, avaliar e definir o limiar para o uso ou não dos dados a serem avaliados (LIMA; SANTOS, 2004)

O presente trabalho propõe o controle da qualidade de dados, por meio do corte de valores extremos, sendo eles mínimos e ou máximos. Esses valores são parametrizados pelo pesquisador, e pela utilização de métodos estatísticos, nos casos onde haja detecção automatizada de falha nas séries de dados, utilizando-se estações de regiões geograficamente próximas (que estejam dentro do raio de distância) também parametrizado pelo pesquisador.

2.1.4 Modelo estatístico para preenchimento de falhas e o controle de qualidade de dados

O preenchimento de dados em séries históricas meteorológicos que possuem falhas, causadas por sensores de estações meteorológicas automáticas ou falhas humanas de anotação, é comumente realizada por métodos estatísticos (FERRARI, 2011).

No trabalho de (FILHO et al., 2013) é apresentado um estudo onde discute os principais métodos de correção de falhas. A partir deste estudo inicial desenvolvido por nosso grupo de pesquisas, esta dissertação utilizará o método denominado regressão linear.

Além do preenchimento temos ainda a questão da qualidade de dados, que passa incialmente pelo pré-processamento, onde as falhas e valores expúrios são localizados e organizandos.

Quanto a qualidade de daos e Preenchimento de Falhas:

1. Controle de Consistência Interna – Valores de mínimos e máximos de chuva, são definidos pelo pesquisado, fazendo com o que o *workflow* avalie se um determinado dados deve ou não ser considerado como válido na série de dados.
2. Controle de Consistência de Tempo e Espaço – O pesquisador parametriza o *workflow*, com o objetivo de definir o raio, para selecionar as estações que podem fornecer dados para o preenchimento de falhas de uma determinada estação, naturalemente o intervalo de tempo, das estações doadoras deve ser compatível com a que recebe os dados.
3. O workflow utiliza a Regrassão Linear (RL) Simples, que considera a existência de uma relação linear entre a séride precipitação Yi, na qual as falhas serão preenchidas (variável dependente), e a precipitação de uma estação selecionada Xi, sendo o modelo descrito como $(Yi = a + b\ Xi)$.

2.2 E-Science

A visão geral de Ciência, sempre ligada à investigação sistemática de eventos e fenômenos baseados em hipóteses, acontecia tradicionalmente nos contextos *in vivo* ou *in vitro*. No entanto com o difusão acelerada da computação vivem-se uma nova fase da ciência que agora acontece *in virtuo* ou *in silico*, que refere-se à mesma investigação e sua sistemática mas agora em ambiente computacional (CRUZ, 2011).

A crescente participação da computação como apoio para as ciências, pode ser confirmada com contribuições em diversos ramos, como por exemplo, na engenharia, onde modelos físicos de alta complexidade, baseados em equações diferenciais são executados em sistemas de supercomputadores, e não somente a execução como a análise e a visualização dos dados gerados. Além desse a adoção dos *workflows* científicos, que são definidos como a especificação de alto nível de um conjunto de tarefas e suas dependências (entre *atividades*), reunidos a fim de alcançar um objetivo específico (DEELMAN et al., 2009).

Na meteorologia, pode-se observar poucos trabalhos que utilizam *workflows* científicos. Entre esse trabalhos destaca-se (ASVIJA B et al., 2010) para implementação do modelos meteorológicos MM5. Nesse trabalho o autor discute sobre as necessidades de oferta de técnicas que permitam computação de massiva, na implementação de modelos climáticos, e que a combinação de computação em grade e *workflows* científico têm muito a oferecer nessa área. Ainda em relação ao trabalho de Asvija, observa que a questão do desempenho computacional é um dos pontos abordados em sua avaliação, sendo esse um dos paralelos traçados com trabalho aqui apresentado.

2.3 *Workflows* Científicos

Os *workflows* científicos são derivados dos artefatos conhecidos por *workflows* concebidos década de 80 para a área de automação de processos. Eles objetivavam a criação, compartilhamento, envio e armazenamento de documentos em uma empresa e ou organização. Como essa tecnologia visava-se a automação de ambientes administrativos. Posteriormente, ela foi adaptada para uso científico nas áreas de Biologia, Química, Física entre outras. Inicialmente foram usados *scripts* para realizar os processamentos científicos. Esses *scripts* eram usados para automatizar parcial ou integralmente processos, mas tinham como ponto negativo a inexistência de suporte a proveniência de dados, além da dificuldade de reaproveitamento de código e dificuldades de desenvolvimento (CRUZ, 2011).

Os *workflows* científicos são uma tecnologia de integração, que fazem uso de diversos recursos computacionais como banco de dados, serviços web, aplicativos locais, servidores e diferentes serviços (TIWARI; SEKHAR, 2007), o que permite a definição de tarefas a serem executadas de forma sequencial ou paralela em um computador ou em ambiente de computação distribuída com o objetivo de agilizar a pesquisa científica sequenciando, determinando e automatizando seus passos. Os workflows científicos possuem dois tipos de representação (abstratos e concretos), nas próximas sub seções apresentaremos suas principais características.

2.3.1 *Workflow* Abstrato

O *workflow* abstrato, se relaciona com os elementos conceituais do processamento e com a etapa de concepção do experimento, onde serão definidos os protocolos pertencentes ao experimento (CRUZ, 2011). Em outras palavras pode-se dizer que o *workflow* abstrato refere-se como os fluxos de dados e seus controles, devem seguir em um experimento, mas em um nível onde os recursos (programas, dados e dispositivos) a serem utilizados não estão materializados. O *workflow* abstrato, é uma representação conceitual das tarefas e do processo como um todo (DEELMAN et al., 2009). Para exemplificar podemos dizer que o *workflow* abstrato está para o pesquisador, como a planta baixa de uma casa para o engenheiro.

2.3.2 *Workflow* Concreto

O *workflow* concreto, é uma instância materializada do *workflow* abstrato, que é codificada e submetida ao Sistema Gerenciador de *Workflows* Científicos (SGWFC) para a efetiva execução do experimento (OLIVEIRA et al., 2012), que nesta fase é o próprio experimento científico do tipo *in silico*. Naturalmente que a execução deste *workflow* concreto ocorre em ambiente computacional, estando, portanto, sujeito a uma série de limitações e condições operacionais. A execução do *workflow* concreto permite ao pesquisador obter de forma automatizada os dados (resultados) oriundos do seu processo científico

Além disso, é possível coletar a proveniência dos dados manipulados pelo experimento e a proveniência da própria construção do *workflow* (DAVIDSON; FREIRE, 2008). A proveniência de dados é abordada em um tópico específico ainda neste capítulo, onde este assunto será tratado com maiores detalhes.

2.4 Sistemas Gerenciadores de *Workflows* Científicos (SGWfC)

Os Sistemas Gerenciadores de *Workflows* Científicos (SGWFC) são sistemas que dão suporte à criação e desenvolvimento de *workflows* científicos do tipo concreto, e que permite ao pesquisador criar *workflows* (abstratos ou concretos) a partir de uma interface gráfica intuitiva. Nessa interface é possível realizar o acoplamento de "caixas" que representam o encadeamento de cada atividade do protocolo do experimento, algumas caixas com funções já definidas e outras prontas para receberem linhas de código, na linguagem apropriada de acordo com o SGWFC. Essa tecnologia permite ao pesquisador sequenciar, documentar e organizar seu experimento *in silico* além de fazer a armazenagem dos dados resultantes de cada experimento. Esse armazenamento pode ser desde arquivos de texto formatados, bancos de dados relacionais, dados abertos ligados na *WEB*, triplas RDF, arquivos XML, entre outros.

Os SGWFC contam com interfaces gráficas intuitivas e de fácil utilização, o que permitem bom grau de produtividade, além de possibilitar aos não programadores o desenvolvimento de seus *workflows*, apesar das interfaces gráficas serem similares as interfaces mais utilizadas nas linguagens de programação. A seguir um exemplo de interface de um SGWFC, conhecido como Taverna (HULL et al., 2006), que é utilizado frequentemente no domínio da Bioinformática.

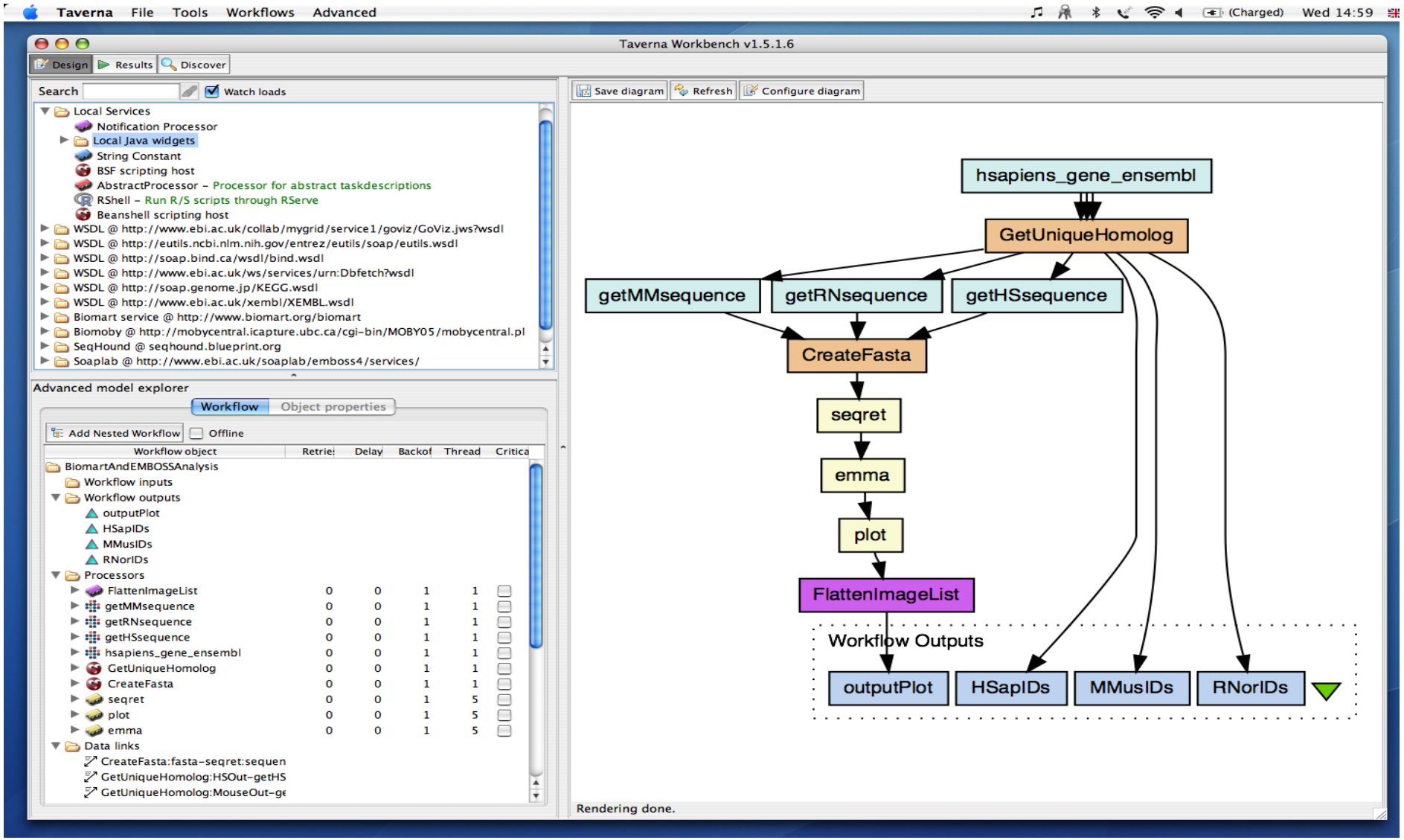

Figura 3 Área de trabalho do SGWFC Taverva

A Figura 3, mostra o ambiente de desenvolvimento do SGWFC Taverna, à esquerda da figura, disponibilizados nos quadrados menores, os serviços e ferramentas disponíveis ao uso do desenvolvedor, já a direita, encontra-se o *workflow* concreto desenvolvido no exemplo.

A utilização de SGWFC pode trazer vantagens ao pesquisador, um exemplo é permitir a análise de resultados parciais em um experimento em tempo de execução. Essa opção permite ao pesquisador optar por interromper um experimento, antes que esse seja finalizado. Esta característica permite um ganho de tempo, quando tratamos de experimentos que envolvem grandes volumes de dados e ou cálculos complexos (MATTOSO et al., 2013).

Outra característica importante dos SGWFC, é o suporte à proveniência, que neste caso permite ao pesquisador, coletar dados relativos ao ambiente de desenvolvimento (versões) ou a execução do experimento, informações de quais usuários participaram das alterações em cada versão (ou quem realizou uma alteração). Acompanhamento da parametrização de um experimento e a versão utilizada no mesmo, essa característica é de suma importância, principalmente quando tratamos da capacidade de reprodução de um experimento. O resultado se torna objeto dotado de um "histórico", que permite ao pesquisador visualizar os passos de um experimento até o seu resultado.

Como parte integrante dessa sub seção de revisão se faz necessário citar os *Workflow Patterns* onde van der Aalst (VAN DER AALST et al., 2002) conceitua o que considera-se como o primeiro passo para o embasamento teórico relacionado ao uso e criação de *workflows* científicos. Neste artigo, tem-se definições de diversos padrões que variam desde simples interações entre as tarefas e suas dependências até laços e controles.

Atualmente existem dezenas de SGWfC, sendo que três SGWC podem ser citados como de interesse para este trabalho, são eles i) Kepler (LUDÄSCHER et al., 2005), ii) Taverna (HULL et al., 2006) e iii) VisTrails (SILVA et al., 2011). Estes são SGWFC de maior destaque no meio científico, e por este motivo recebem um maior detalhamento (CRUZ, 2011). Estes SGWFC são de interesse pela presente pesquisa por possuírem algumas características necessária à solução proposta, como por exemplo, versionamento, suporte à proveniência, execução remota e adaptação do *workflow* concreto, e a utilização de web services, serviços em nuvens e clusteres de HPC.

2.4.1 *Kepler*

O sistema *Kepler* é bastante tradicional na área de *E-Science*, foi desenvolvido com base em Java (com código aberto), e oferece suporte a criação de *workflows* utilizando linguagem MoML, que é derivada do XML, utiliza se também do conceito diretor/ator para representar os componentes do *workflow* concreto e definir o tipo de comunicação entre eles. O diretor controla os atores, sendo os atores os responsáveis pelo processamento dos dados disponíveis em suas entradas. Seu suporte à computação distribuída ainda encontra se em desenvolvimento. Outro fato importante é a inexistência de interface de desenvolvimento de novos atores ("The Kepler Project — Kepler", 2014).

De maneira geral, o sistema *Kepler* possui os requisitos essenciais para um SGWFC, um deles é a capacidade de acesso a recursos e serviços remotos, que inclui *web services*, o que permite inclusive a instanciação de um determinado serviço web como um Ator do processo (*workflow*). Outra característica são as extensões que permitem o uso de *Grid Computing* (LUDÄSCHER et al., 2005) e mais recentemente existem atores que suportam conexão com o ambiente de nuvem. Até o momento, não se verificaram atores dedicados ao processamento em ambiente de GPU.

2.4.2 Taverna

Este sistema foi desenvolvido pelo projeto *MyGrid*, o Taverna é uma ferramenta de composição e execução de *workflows* concretos, que se utiliza do motor *Freefluo*. Idealizado principalmente para problema de domínio da bioinformática, consiste na coleção de processadores além de *links* de controle, seus processadores podem conter múltiplas entradas e ou saídas de dados, onde os *links* estabelecem o controle entre as dependências dessas entradas e saídas de dados ("myGrid", 2008).

Com processadores implementados em Java, é possível realizar a execução do experimento de forma local ou remota. As saídas de dados são na sua maior parte de forma textual. No entanto é possível usar plug-ins que permitem a visualização e coleta de descritores de proveniência. Usa a linguagem conceitual Scufl para a descrição dos *workflows*, que é baseada no XML (TAN et al., 2009).

O Taverna possui ainda amplo suporte a *Web Services*, é fornecido como sistema de código aberto, e foi idealizado por seus criadores principalmente para criação de *workflows* na área de Bioinformática (TAN et al., 2009)

2.4.3 *VisTrails*

O *VisTrails*, é um sistema modular de gerência de proveniência e *workflows* científicos desenvolvido em *Python* (com código aberto), fornece amplo suporte a exploração e visualização de dados, permite a coleta de proveniência prospectiva, o motor de execução é o *VisTrails Cache Manage*r. ("VisTrails Documentation", 2013) .

Criado sob a perspectiva de ser um sistema capaz de coletar diferentes tipos de proveniência e oferecer a visualização de dados, o *VisTrails* é um dos SGWFC, é um ambiente simples e voltado desde avançadas pesquisas científicas até a execução de aulas com o objetivo de gerar conhecimento na área de visualização de dados científicos (SILVA et al., 2011).

Como os demais SGWFC, o *VisTrails* oferece suporte a *Web Services*, traz suporte nativo, à proveniência além de permitir com facilidade, a adaptação de *workflows* concretos, por meio de código *Python* (FREIRE et al., 2008). Nos próximos capítulos desse trabalho, serão

apresentados maiores detalhes sobre o *VisTrails*, sendo essa a plataforma escolhida na condução dessa pesquisa.

2.4.4 Matriz de Decisão

A partir da observação e análise técnica das principais características dos SGWFC listados, foi realizada a escolha e posterior definição do sistema a ser usado neste trabalho, elaborou-se uma matriz de decisão que fosse capaz de pontuar as principais características dessas ferramentas, através desta matriz podem ser atribuídas pesos que varia em três valores a serem assumidos por cada um dos quesitos, Os valores correspondem ao nível de suporte dado pelo SGWFC no quesito avaliado. Os valores, 1 – para presença do quesito, 2 – para presença aperfeiçoada do quesito e 3 – presença sem necessidade de adaptações para o projeto.

Tabela 1- Matriz de decisão.

Critério	SGWFC		
	Kepler	**Taverna**	**Vistrails**
Suporte a proveniência	0	2	3
Adaptação de workflow concreto	3	2	2
Suporte a versionamento	0	0	1
Suporte a execução remota	1	1	1
Total	4	5	7

De acordo com a matriz da Tabela 1, o SGWFC *VisTrails*, apresentou maiores vantagens técnicas do ponto de vista deste trabalho, por apresentar principalmente as seguintes características:

Apoio a coleta de proveniência retrospectiva e prospectiva, aumenta a possibilidade de desenvolvimento de novos módulos baseados nas ferramentas de adaptação para o *workflow,* como exemplo a ferramenta python source, que permite a criação de módulos usando código python. Além disso, o *VisTrails* realiza automaticamente o controle de versão do *workflow*, o que permite inclusive a execução de versões anteriores no mesmo ambiente de desenvolvimento. Baseados nestes motivos, o SGWfC que será adotado nesta dissertação é o *VisTrails* pois apresenta mais vantagens que os demais.

2.5 Proveniência

A Proveniência é defina pelo dicionário de *Oxford,* como a fonte ou origem de um objeto; isto é o histórico ou *pedigree*, o registro das últimas alterações e a passagem por locais ou proprietários. A Proveniência é amplamente usada em diversas áreas do conhecimento humano, como exemplo as Belas Artes, usa a proveniência como a forma de rastreio para localizações, propriedade e originalidade de peças de grande importância e valor comercial, o que visa garantir aos compradores a originalidade e conhecimentos das obras em questão (CRUZ, 2011). Na computação, a proveniência de dados visa acompanhar as alterações ocorridas em um objeto digital, dessa forma pode-se dizer que para entender, interpretar e sequenciar as modificações ocorridas nos dados até um determinado resultado, deve-se contar com a proveniência de dados (FREIRE et al., 2008).

Em se tratando de experimentos científicos, a proveniência ajuda na análise e interpretação de resultados; pelo exame da sequência de passos que levou o experimento a um resultado. Dessa forma o pesquisado pode entender melhor a cadeia de etapas de um

experimento, o que permite inclusive que através dos dados de entrada, que um experimento seja reproduzido com maior facilidade e fidelidade (FREIRE et al., 2008).

Comumente cientista e engenheiros realizam grandes esforços e tempo, armazenando e gerenciando dados, com o objetivo de saber: Por quem e quando um conjunto de dados foi gerado? Que processo gerou determinando conjunto de dados? Quem e quando modificou um conjunto de dados? Todo esse esforço se deve a dificuldade de obter essas respostas de forma manual (FREIRE et al., 2008).

Pode-se detalhar as duas formas de proveniência que podem, ser coletadas nos experimentos científicos. A Proveniência Prospectiva, ligada aos metadados relativos as sequencias de tarefas (operações) computacionais a serem modeladas, como por exemplo, se determinado passo será representado por um *script* ou atividade de *workflow*, a sua ordem, e ainda pode acompanhar dados como a versão e outros detalhes. Por isso essa proveniência corresponde aos passos a serem seguidos para se alcançar um dado ou conjunto de resultados (DAVIDSON; FREIRE, 2008; FREIRE et al., 2008).

Já a Proveniência Retrospectiva, coleta metadados relativos aos passos executados pelo *workflow*, assim como os detalhes do ambiente que cerca essa execução. Preocupa-se também com o conjunto de dados gerados a partir desta execução, fazendo assim a relação entre eles, ambiente X resultados (dados)(FREIRE et al., 2008).

A Proveniência de dados coletada a partir da utilização de *workflows* científicos pode ser usada com diversos objetivos dependendo do problema que se busca solucionar. Neste trabalho, a proveniência de dados têm como objetivo o acompanhamento da qualidade dos itens de dados meteorológicos, o que permite ao pesquisador acompanhar cada passo entre o pré-processamento e a obtenção do resultado final, que é a aplicação dos métodos estatísticos aplicados as séries de dados. Outra fonte de proveniência é o acompanhamento das etapas de desenvolvimento do próprio *workflow* científico.

2.5.1 Metamodelo de Proveniência – PROV

O PROV, é um metamodelo originado como produto de um grupo de trabalho, organizado pelo W3C (*World Wide Web Consortium*), com a finalidade de criar uma família de documentos, que sirvam de mapa ou roteiro, para o desenvolvimento da modelagem de dados apoiado em proveniência e intercambio desses dados via *web*. A família de documentos que formam o PROV são:

Tabela 2:Orientaçõe modelo PROV-DM

PROV-DM: orientações para modelagem de dados, define os tipos de dados e suas relações (*The PROV Data Model*).
PROV-O: guia de ontologia, baseado OWL2 *Web Ontology Language*i, visa descrição de dados para o intercâmbio entre sistemas heterogêneos (*The PROV Ontology*).
PROV-N: fornece exemplos de modelagem PROV, afim de facilitar a aprendizagem da modelagem de dados, por parte dos desenvolvedores no PROV-DM, utilizando semânticas formais do PROV Prov (*The Provenance Notation*).
PROV-AQ: trata a forma como os protocolos da Internet, devem ser usados para o acesso e recuperação de dados de proveniência (*Provenance Access and Query*).
PROV-XML: orientações gerais para exportação de dados, modelados a partir do Prov-DM, para o padrão XML (*The PROV XML Schema*).
PROV-Dictionary: descreve a estrutura PROV, visando facilitar a criação de dicionário de dados (*Modeling Provenance for Dictionary Data Structures*)

Tabela 2. Continuação

PROV-Links: Apresenta aos relacionamentos entre objetos (entidades), que estão em pacotes separados (descrição de dados, em um BD de terceiros) *(Linking Across Provenance Bundles*).
PROV-DC: mapeamento parcial do termos *Dublin Core* (esquema de metadados, para descrição de objetos virtuais), para o PROV-O e OWL2 (*Dublin Core to PROV Mapping)*
O Prov-DM, (*Provenance Data Model*) é o padrão desenvolvido pela W3C, para nortear a modelagem de dados apoiado em proveniência, visando a ampla publicação e o intercâmbio de dados via Internet (“*PROV-DM: The PROV Data Model”,* 2013).

Estas especificações permitem ao desenvolvedor criar as relações entre seus dados de proveniência, tendo como base três papeis principais, sendo eles: entidades, atividades e pessoas envolvidas (*entities*, *activities*, e *people involved*).

O PROV- DM é desenvolvido de acordo com as oito recomendações gerais do W3C *Provenance Incubator Group*, que são: Identificação dos objetos, por meio das classes, Ator (personagem), Entidade ou Processos (etapas). Acesso a informações relacionadas em outros padrões. Acesso a proveniência. Geração de dados de proveniência da proveniência, Reprodutibilidade, Versionamento, Representação de conhecimento.

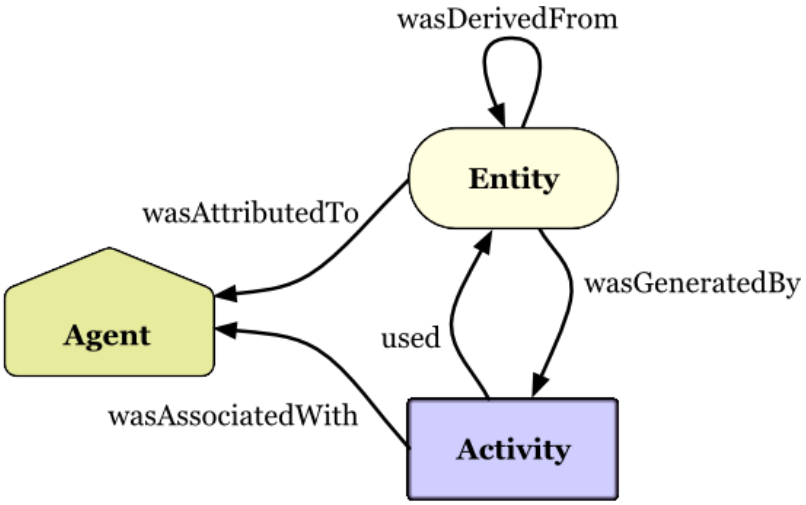

Figura 4- Relação agente, entidade e atividade.

A figura 4, é um exemplo de organização da proveniência no metamodelo PROV sob a forma de um grafo. Nela se observa a relação entre entidade, atividade e agente, onde são apresentados as ações (sequencias) e a forma como cada um desses papeis interagem, exemplo: entidade (entity) X --- Atribuída → por Agente(agente) Y –Criada→ pelo evento (activity) Z.

Essas instruções servem diretamente para criação de arquivos de exportação, como XML ou no formato Turtle (“PROV-DM: The PROV Data Model”, 2013), que são os padrões sugeridos pelo PROV para exportação de dados via *web*.

2.6 Processamento em Larga Escala

Experimentos científicos no contexto de *e-Science* podem manipular grande volume de dados em cada execução do experimento, sendo necessário reexecutar vários *workflows* e programas que requerem quantidade significativa de recursos computacionais, além do uso diversificado de dados. Com isso, é natural a associação de *e-science* à computação de alto desempenho (HORTA et al., 2013a), que geralmente pode ser distribuída ou paralela.

2.6.1 Processamento Distribuído

Uma forma direta de se definir os sistemas distribuídos é: "Um sistema distribuído é uma coleção de computadores, que ao usuário se apresenta como um único e coerente sistema" (TANENBAUM; STEEN, 2007). Isto é a associação de diversos computadores, conectados por uma rede, com um alvo de processamento comum, onde cada computador (nó), possui uma tarefa específica (ou parte de uma) a ser processada a fim de se alcançar esse objetivo.

No entanto, até o advento dos sistemas distribuídos foi necessária a evolução dos microcomputadores, além da criação das redes locais de computadores. O que permitiu a comunicação de centenas de computadores. A partir desses eventos foi possível iniciar o processo de distribuição de tarefas entre computadores (TANENBAUM; STEEN, 2007).

Sistemas distribuídos envolvem ainda o conceito de transparência, pois, aos seus usuários não deve ser perceptível a forma como os nós estão dispostos na sua organização geográfica e tecnológica, ou ainda a forma como os recursos são disponibilizados. Também fica a critério do sistema os mecanismos de recuperação de falhas e a concorrência de recursos (TANENBAUM; STEEN, 2007).

Dentro dos sistemas distribuídos se faz necessário citar seus tipos principais, sendo eles: (i) *Cluster*, onde são usados computadores idênticos, sendo possível inclusive usar computadores pessoais do tipo workstation. Este tipo chama a atenção pelo custo/benefício de construção; (ii) *Grid* computing que se caracteriza principalmente pela grande heterogeneidade, pois concentra recursos de diversas entidades, criando uma espécie de entidade virtual (TANENBAUM; STEEN, 2007); temos ainda a iii) *Cloud Computing* que é a convergência das várias tecnologias e conceitos de sistemas distribuídos (TAURION, 2009). Os conceitos de Grid e Cloud não serão abordados neste trabalho.

Em termos de *e-Science*, os sistemas distribuídos são de suma importância, pois é a partir desses sistemas, que se torna possível a utilização de grande poder computação e a processamento de grandes volumes de dados(HEY; TOLLE, 2009).

Nesta pesquisa busca-se a combinação dos aspectos citados, com a execução de *workflows* científicos.

Quando nos referimos à computação de alto desempenho no contexto do processamento científico, ela se mostra amplamente aplicável na solução de problemas que envolvem grandes volumes de dados para a área de Meteorologia. No entanto, é necessário detalhar as tecnologias utilizadas pois temos a disposição várias tecnologias tais como: redes heterogêneas de computadores, que se utilizam dos computadores interligados em rede (com diferentes tipos de processadores), baseados em um protocolo de troca de mensagens como *Message Passing Interface* (MPI) ("Open MPI: Open Source High Performance Computing", 2014) ou ainda o uso do *Open Multi-processing* (OpenMP) ("OpenMP.org", 2014) nas arquiteturas baseadas em processadores de múltiplos núcleos (JACOB, 2010; LASTOVETSKY, 2002). Existem outros tópicos importantes relativos a computação paralela,

por exemplo, "Memória compartilhada *vs* Memória Distribuída", SIMD (*Single Instrucution Multiple Data*) vs MIMD (*Multiple Instruction Multiple Data* (DENNING; TICHY, 1990).

2.6.2 Processamento paralelo

Neste trabalho, o modelo de computação que serve como base teórica de fundamentação é o processamento paralelo. Ele consiste na utilização de um ambiente computacional capaz de realizar diversas etapas de um evento simultaneamente. Isto requer um ambiente que permita a execução de partes de um algoritmo de forma paralela. Desde que o problema computacional a ser abordado permita "separação" de suas etapas (BARBOSA, 1996). Este tipo de processamento requer *hardware* apropriado, além de sistemas operacionais, linguagens de programação e ambientes computacionais desenvolvidos para esse fim. Neste modelo de processamento destaca-se o ganho de desempenho, porém ressalta-se a existência de dificuldades de implementação.

Neste trabalho temos como foco o uso da computação paralela baseada em *Graphic Process Unit* (GPU) que é um tipo especial de processador, construído especificamente para o processamento de gráficos em 3D (renderização em especial). O desenvolvimento e uso desse tipo de processador cresceu muito nos últimos anos, ao ponto de se perceber que o poder computacional disponível para jogos era potencialmente aplicável para resolução de problemas científicos (CHAKRABARTI et al., 2012).

2.6.2.1 Unidade gráfica de processamento (GPGPU)

GPGPU é o termo utilizado para definir a programação em GPU com um fim diferente de processamento gráfico(HARRIS, 2002). A partir da definição, é possível perceber que a computação paralela baseada em GPU´s torna-se uma das principais estratégias para se resolver problemas científicos que demandam grande esforço computacional *(HONG*; DA-FANG; XIA-AN, 2012).

2.6.2.2 CUDA

Criada pela NVIDIA, a arquitetura CUDA permite a utilização de GPGPU´s em computação paralela. Esta arquitetura chama a atenção por permitir o compartilhamento e o alto fluxo de dados em memória, ela utiliza o formato de múltiplos dados com uma tarefa (SIMD)(CHAKRABARTI et al., 2012).

Concebida para escrita e execução de programas de propósito geral, a arquitetura CUDA permite o desenvolvimento de programas em computação paralela através da criação de *threads*. Pode-se definir como *thread* uma instancia de *kernel*, isto é um programa que é executado dentro da GPU. No entanto não é necessário ter conhecimentos de processamento de gráficos (OKITSU; INO; HAGIHARA, 2010).

Para a escrita desses programas é possível utilizar linguagens já conhecidas como o C e FORTRAN que são amplamente suportadas pelo CUDA. Além da incorporação do compilador CUDA, conhecido como NVCC, em interfaces de integradas de desenvolvimento como Eclipse e Visual Studio ("CUDA Toolkit Documentation", 2013).

Analisando as características do CUDA percebe-se seu potencial na área de ensino e pesquisa científica pois abstrai as questões relacionadas a alocação de recursos de *Hardware*, para permitir a execução de algoritmos paralelos, ficando como foco ao pesquisador a criação de um código funcional e eficiente no que diz ao algoritmo que envolve o problema

(NICKOLLS et al., 2008), sendo sem dúvida essa uma das características necessárias para o desenvolvimento da presente dissertação.

2.6.3 Arquitetura multinúcleos

A arquitetura multinúcleos, baseia-se na execução de mais de um processo por processador, sendo esse segundo processo uma *thread* que dependendo da tecnologia do processador pode ser uma ou mais *threads* por núcleo. Na figura 5 é apresentado um esquema conceitual de dois processadores, um que permite a execução de uma *thread* por núcleo, e um segundo onde cada processador pode executar duas *threads* por núcleo, nesse caso essa tecnologia é conhecida como como *hyper-threading* (HONG; DA-FANG; XIA-AN, 2012). Basicamente dessa forma se organiza a arquitetura multinúcleos onde núcleos de processador são adicionadas e dependendo da tecnologia cada núcleo pode receber um número *n* de processos. Com o avanço da tecnologia de miniaturização hoje é possível encontrar processadores com seis núcleos com suporte a doze *threads*, isso apenas para os computadores pessoais.

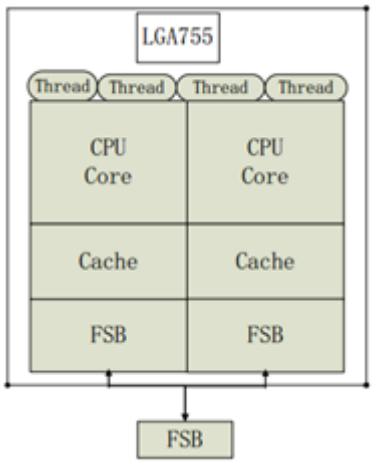

Figura 5 Arquitetura multi núcleos (Fonte: Hong, Da-fang e Xia-an, 2012)

2.6.4 Arquitetura GPGPU

Em se tratando da arquitetura GPGPU, pode-se observar que o número de *threads* suportadas são muito maiores, por conta do número de núcleos GPGPU que por serem mais simples são inseridos diretamente na placa GPU. Além do elevado número de núcleos essa arquitetura demonstra o uso de múltiplas matrizes de processadores, essas matrizes é que permitem a essa arquitetura a cria um número considerável de *threads* (que depende da tecnologia embarcada na placa) forma uma rede onde comunicam-se e compartilham memória (HONG; DA-FANG; XIA-AN, 2012). Conforme a Figura 6, destacam-se a SPA (*scalable stream processor array*), que é a matriz de processadores e seu conjunto de memórias interconectadas, que juntos formam o sistema de criação e controle de múltiplas *threads*.

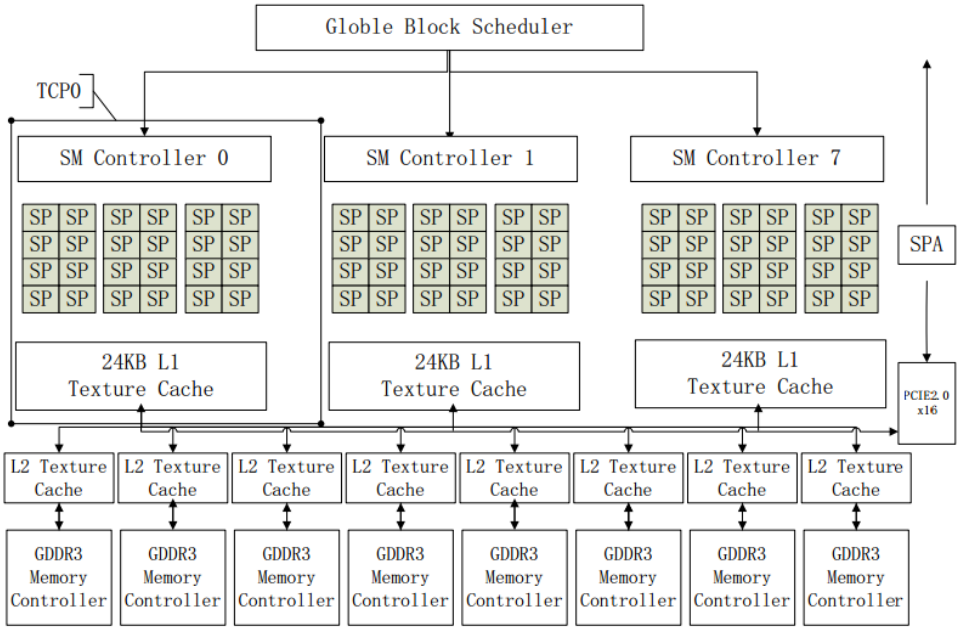

Figura 6 - Arquitetura GPGPU (Fonte: Hong, Da-fang e Xia-an, 2012)

2.6.5 Arquitetura GPGPU vs arquitetura multinúcleos

Apesar de ambas arquiteturas serem destinadas ao paralelismo, GPGPU e *multi* núcleos possuem características particulares, segundo (HONG; DA-FANG; XIA-AN, 2012), o processamento baseado em GPGPU, é um nível mais fino de processamento, capaz de lidar muito bem com cálculos em larga mas escala de complexidade simples. E quanto o *multi*núcleos é adequado para lógica complexa com processamento em menor escala.

Com isso percebemos que cada *thread* GPU apesar de um poder computacional menor, por estar disponível massivamente, pode em tese superar resultados das poderosas threads da arquitetura multinúcleos.

2.7 Trabalhos Relacionados

Atualmente, tem-se à disposição trabalhos na área de pré-processamento de dados meteorológicos que incorporam somente parte dos objetivos e métodos propostos nesse trabalho. A começar pelo trabalho de (FILHO et al., 2013) que propõe um sistema de pré-processamento de dados meteorológicos em uma plataforma *Web*, mas que, no entanto não traz uma solução ou aplicação paralela de alto desempenho. Temos ainda (MAGINA, 2007) que propõe formatos de tratamento estatísticos para series históricas meteorológicas, mas conta somente com macros em planilhas MS Excel para aplicação desse tratamento, o que torna o trabalho humano massivo e sujeito a falhas. Além disso, este trabalho não incorpora as questões de proveniência de dados.

No que tange à computação massiva e paralela, temos o trabalho de (HORTA et al., 2013b), que fala sobre os caminhos do uso de *workflows* científicos usando computação paralela, mas que não apresenta uma solução desenvolvida que se aplique ao problema que norteia este trabalho, ficando somente no campo da investigação do que é possível realizar juntando *workflow* científico, computação massiva paralela.

Já (ASVIJA B et al., 2010, p. 5) sugere o uso de *workflow* científicos para implementar modelo meteorológico MM5, que trata da predição ou simulação em meteorologia de meso

escala ou escala regional de fenômenos atmosféricos, como brisas, *thunderstorm convection*, não contemplando tratamento de dados e coleta de proveniências.

Dessa forma, verifica-se a necessidade e importância do desenvolvimento da combinação de pesquisa, metodologia e tecnologias propostas neste trabalho, pois essas características combinadas possuem capacidade de beneficiar os pesquisadores da área de meteorologia e dessa forma, permitindo a esses uma ferramenta de pré-processamento de dados que obtenha dados curados em larga escala de forma autônoma, baseada em *e-science*.

Tabela 3 - Comparação entre trabalhos relacionados.

Características	**Autores**				
	Abordagem Proposta	**Filho *et al*, 2013**	**Magina, 2007**	**(Horta, Silva, *et al*, 2013**	**Asvija B e*t al*, 2010**
Worflows Científico	X			X	X
Coleta de Proviniencia	X	X			
Qualidade e Preenchimento de Dados	X	X	X		X
Computação Paralela	X			X	
cUDA	X				

A partir da Tabela 3, é possível perceber que etse trabalho, fornece uma combinação que pretende preencher a lacuna deixada por trabalho relacionados, fazendo com que a visão *e-Science* (*workflows* científicos e coleta de proveniências), computação paralela (tecnologia baseada em GPU), sejam um diferencial na área de meteorologia, mais precisamente na hidrologia.

A hidrologia, é uma área que requer dados de qualidade, e que precisa de grande capacidade computacional, tendo em vista as diferentes fontes de dados heterogêneos, o que compões o seu grande volume, e a realidade que envolve esses dados, que apesar da grande disponibilidade, por muitas vezes estão degradados e sem nenhum tipo de tratamento.

Ainda com base na Tabela 3, percebe-se que os trabalhos relacionados são voltados para atender áreas especificas da computação ou da meteorologia, sem aplicação de novos conceitos científicos ou de tecnologias adequadas para o tratamento desses dados e da sua proveniência.

Dessa forma, pode-se que considerar que este trabalho, adequa as necessidades da meteorologia, com a visão *e-Science* e ainda promove a utilização de tecnologias inovadoras na área de computação.

3 ABORDAGEM PARA TRATAMENTO DE DADOS METEOROLÓGICOS APOIADOS POR *WORKFLOWS* CIENTÍFICOS

Este capítulo tem em vista apresentar a solução computacional adotada para automatizar o tratamento das longas séries históricas de dados meteorológicos, mais especificamente de dados pluviométricos, a partir do desenvolvimento e execução de *workflows* científicos em ambientes de computação paralela do tipo GPU. Os *workflows* desenvolvidos representam a especificação formal de um protocolo científico comum ao domínio da Meteorologia e que representa os passos a serem executados em um determinado experimento científico (DEELMAN et al., 2009).

Neste capítulo, são apresentados os *workflows* utilizados, assim como os módulos de processamento paralelo que exploram o paralelismo das placas GPUs. Além disso, apresenta-se a aplicação de métodos estatísticos do tipo regressão linear adotado nesta pesquisa.

Além das especificações citadas, do envolvimento da computação distribuída e paralela, são apresentados os bancos de dados para armazenar os dados brutos, curados e os descritores de proveniência dos experimentos. Tais itens serão melhor detalhados nas próximas subseções.

3.1 Abordagem Apoiada em *Workflows*

O *workflow* abstrato tem como objetivo representar com alto nível de abstração, os processos e as interações entre as etapas de um experimento e suas dependências. No entanto, antes da modelagem abstrata do *workflow*, tem-se um diagrama que representa o processo realizado manualmente pelos Meteorologistas, durante o processo de apuração e correção de falhas nas séries históricas.

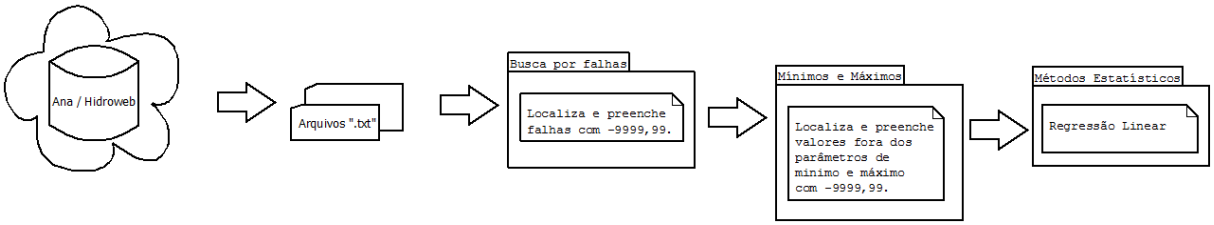

Figura 7- Passos manuais para detecção correção de falhas.

Observa-se que o processo representado na Figura 7 está centrado no controle manual executado por parte dos pesquisadores. O processo se inicia pelo acesso ao sistema Hidroweb da Agência Nacional de Águas (ANA), com o download arquivos de dados pluviométricos. Em seguida, pesquisador localiza e preenche manualmente as falhas e adota o código internacional "-9999.99". Em seguida o sistema busca por valores mínimos e máximo extremos. Caso esses valores fiquem fora do limite físico, também receberão o código "-9999.99". Por fim, se aplica o método estatístico para geração da médias mensais de cada uma das estações (regiões) de interesse. Esses passos são realizados de forma manual e seqüencial, e por isso, são capazes de consumir tempo significativo, gerando trabalho demasiado, pois o pesquisador (ou equipe) dedica muito esforço e tempo ao pré-processamento dos dados em detrimento de analisar e gerar resultados científicos a partir dos dados obtidos a partir do sistema Hidroweb da ANA.

A Figura 7 também descreve das atividades abstratas do *workflow* adotado como solução desta proposta, começando pela representação do banco de dados da ANA, onde são

armazenados os arquivos em que o pesquisador, através do sistema Hidroweb, seleciona as estações (regiões), nas quais deseja realizar o processo de correção de falhas, e pelo sistema, faz o *download* dos arquivos. Esses arquivos são do tipo "*.txt", e serão inteiramente lidos e processados pelos *workflows* concretos responsáveis por essas etapas.

3.1.1 *Workflow* Abstrato

No processamento automatizado, utilizando *workflows*, temos a interação entre o pesquisador e a solução computacional proposta nesta dissertação, por meio de uma interface gráfica, que solicita ao pesquisador, parâmetros de execução, sobre o experimentoa ser realizado, como a localização e quantidade de arquivos, neste caso, referentes aos arquivos de interesse que serão recuperados do Hidroweb. Além disso, serão informados quais são os parâmetros para avaliação dos dados contidos nos arquivos, tais como o valor de mínimo e máximo, que são de grande importância, pois valores fora dos assinalados pelo pesquisador serão considerados como dados inválidos e serão marcados como -9999.99.

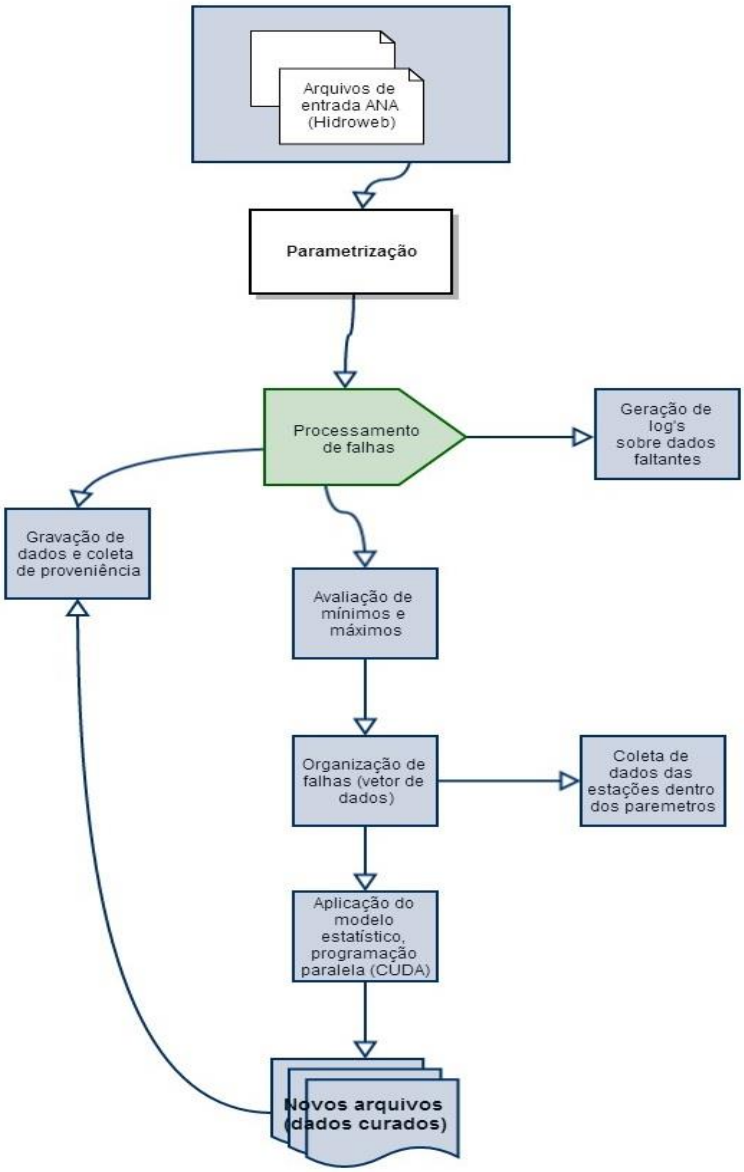

Figura 8 - *Workflow* abstrato (proposta de automatização)

Em seguida, ocorre a leitura dos arquivos, sendo um arquivo por estação, o número de estações é definido pelo conjunto de estações selecionado pelo pesquisador, e seu número é definido na etáá de parametrização do *workflows*, em seuiguida o *workflow*, realiza a busca por falhas nas séries, essas falhas se caracterizam pela ausência de valores na série histórica, isto é, em um arquivo do tipo ".txt", separado por pontos e vírgulas, naturalmente deve conter um valor numérico entre cada par consecutivo de pontos e vírgulas. As falhas são caracterizadas pela ausência desses valores. A Figura 9 apresenta uma falha em um arquivo de série histórica.

A partir da definição, pode-se dizer que a parte sequencial do *workflow* se encarregará pela busca de falhas e deve percorrer por completo todo o arquivo de cada uma das estações, identificando as falhas e marcando cada uma dessas falhas com -9999.99. Com isso, serão identificadas as datas onde os dados meteorológicos não foram coletados de maneira correta, o que vai permitir em uma etapa futura a aplicação de um ou mais métodos estatísticos para o preenchimento dessas falhas.

```
02241004;1;01/08/1967;1;;;;0;0;0;;0;;;;;;;;;;;;;;;;;;;;;;0,2;;;;;;;;;;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
02241004;1;01/09/1967;1;;;;0;0;0;;0;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;
02241004;1;01/10/1967;1;;;;0;0;0;;0;;;;;;;;;;;;;;;;;;;0,2;0,2;;;;;;;;;;0,2;0,2;0;0;0;0;0;0;0;0;0;0;0;0;0;
```

Figura 9 - Falha em série histórica

Naturalmente que os dados coletados são de intervalo diário, no entanto, a aplicação da correção não será feita nas unidades diárias e sim na média mensal de cada estação que contiver falhas em sua série.

Em seguida, temos a análise dos valores de corte. Estes valores são os mínimos e máximos que podem ser alcançados pelos valores das séries. Valores fora desses limites parametrizados pelo pesquisador na fase de interação (módulo de parametrização) devem também ser substituídos por -9999.99. O que não significa que os dados relativos a esses valores, são de fato inválidos, mas também podem significar que, naquela "rodada" do experimento, os valores de interesse devem estar compreendidos no intervalo determinado pelo pesquisador.

Ainda em relação ao *workflow* abstrato representado na Figura 9, destaca-se o uso da computação paralela. Na etapa ocorre a aplicação de métodos estatísticos para a correção das falhas dos dados meteorológicos brutos. O *workflow* separa e processa os dados referentes às estações com falhas nas séries de dados e as estações que possam servir como "pontos de correlação" de dados para aplicação dos métodos estatísticos, que são as estações, dentro do raio parametrizado, pelo pesquisado. Ressalta-se que nesta pesquisa, o parâmetro que define se uma estação pode ou não doar dados para esse processo, é a distância em quilômetros entre a estação que sua série com falhas e sua circunvizinhança. Por isso na fase de interação (módulo de parametrização), o *workflow* concreto é parametrizado pelo pesquisador que informa o raio, que definirá as estações que terão seus dados usados no processo de correção de falhas.

Um dos objetivos desta pesquis é produzir módulos do *workflow* baseados em computação paralela utilizando a tecnologia GPU. Neste sentido, considera-se a seguinte abordagem para o processamento paralelo.

De maneira geral, quando trata-se de uma série de dados, se torna um padrão a criação de vetores para a realização de cálculos, e o problema em questão não é uma exceção. Cada série de dados será tratada como um vetor e serão realizadas as operações desejadas e comparações caso necessário. O que muda quando trata-se de computação paralela é que os dados de cada vetor não serão selecionados e tratados por um contador que indexa os "endereços" de cada posição para aplicar as operações, mais sim pela criação de um espaço de memória na GPU, que recebera os vetores e realizará a operação desejada em lote.

3.1.2 Integração w*orkflow* vs programa CUDA

Um dos desafios dessa pesquisa foi a integração de um *workflow* desenvolvido em *VisTrails* e a execução do modelo estatístico usando recursos de paralelismo, neste caso CUDA, uma vez que o *VisTrails* não possui uma ferramenta específica para o uso de paralelismo em CUDA.

Com isso, foi necessário criar um sub-*workflow* de integração em código *python*, usando o componente "*python source"* do *VisTrails*, tanto para a execução local quanto para a execução em uma máquina remota. Sendo a versão remota baseada na execução de comando utilizando SSH (*Secure Shell*).

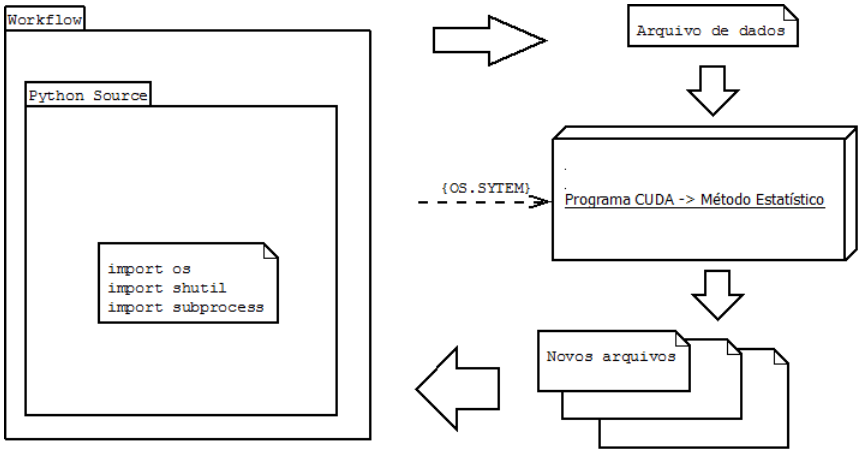

Figura 10 - Integração *workflows* e programas CUDA

Conforme representado na Figura 10, pode-se dizer que o *workflow*, a partir do componente *python source*, utilizando a bibliotecas de chamada de processos via sistema operacional, cria um sub processo, que por sua vez é a execução do programa em CUDA, que usa como entrada de dados os arquivos dados criados pelo *workflow* nas etapas anteriores. Naturalmente que, após a aplicação do métodos estatísticos são gerados novos arquivos, que por sua vez servem de entrada para o *workflow*, pois contém dados curados, que servem como resultado e fazem parte também da coleta de proveniência dos dados do experimento.

O *workflow* responsável aplicação do método estatístico, conta com o auxílio de um outro *workflow* que envia os dados selecionados a uma máquina paralela remota (Coyote), que recebe as séries de dados, e que de fato, aplica os cálculos sobre essas séries. Este *workflow* conta com comunicação SSH (*Secure Shell*), que inicia e controla uma sessão de comunicação entre o computador local, onde o *workflow* está sendo executado e a máquina remota. Nesta sessão, é executado o programa que contém o código CUDA, que aplica a regressão linear de forma paralela. No capítulo seguinte, avalia-se os resultados relativos a aplicação de paralelismo, como descrevendo a aplicação e seu desenvolvimento.

3.1.3 Programação paralela em CUDA

Um dos objetivos deste trabalho é utilizar a construção de um programa baseado em computação paralela usando CUDA. Nesta sessão serão demonstradas as principais diferenças entre a computação sequencial e paralela no problema proposto nesse trabalho.

De maneira geral, quando tratamos de uma série de dados, se torna um padrão a criação de vetores para a realização de cálculos, e o problema em questão não é uma exceção, cada série de dados será tratada como um vetor e serão realizadas as operações desejadas e comparações, caso necessário. O que muda quando tratamos de computação paralela é que os dados de cada vetor não serão selecionados e tratados por um contador que indexa os "endereços" de cada posição para aplicar as operações, mas sim pela criação de um espaço de memória na GPU, que receberá os vetores e realizará a operação desejada em lote.

A Figura 11 mostra o comportamento de um programa que soma dois vetores e grava o resultado da soma desses vetores em um terceiro vetor, contando com a ajuda de um contador para que o resultado de cada soma vA[i] + vB[i], seja armazenada em vC[i].

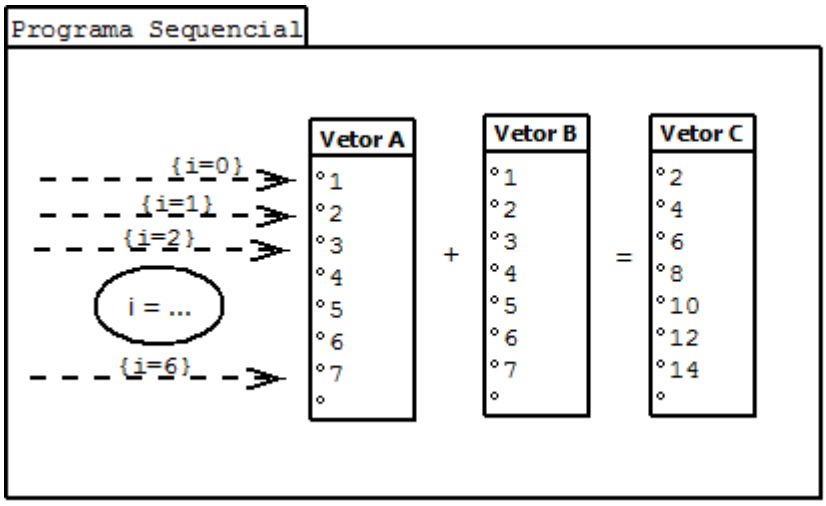

Figura 11 - Soma de vetores sequencial.

Na Figura 12 tem-se a versão paralela do mesmo programa, desenvolvido com base em CUDA, onde o processo é realizado de forma a oferecer ganho de desempenho, pois não se utiliza de um contador para indexar os vetores, mas sim, reserva um espaço de memória na GPU (*device memory*), os vetores serão alocados e executados em uma bloco de *threads*, são controlados pela GPU, uma vez implantados utilizam a função que controla a operação a ser realizada entre os vetores e as realiza em todo o bloco, ficando cada uma das tuplas dentro de uma *thread*, que são as células no interior do bloco, que são executadas de uma só vez.

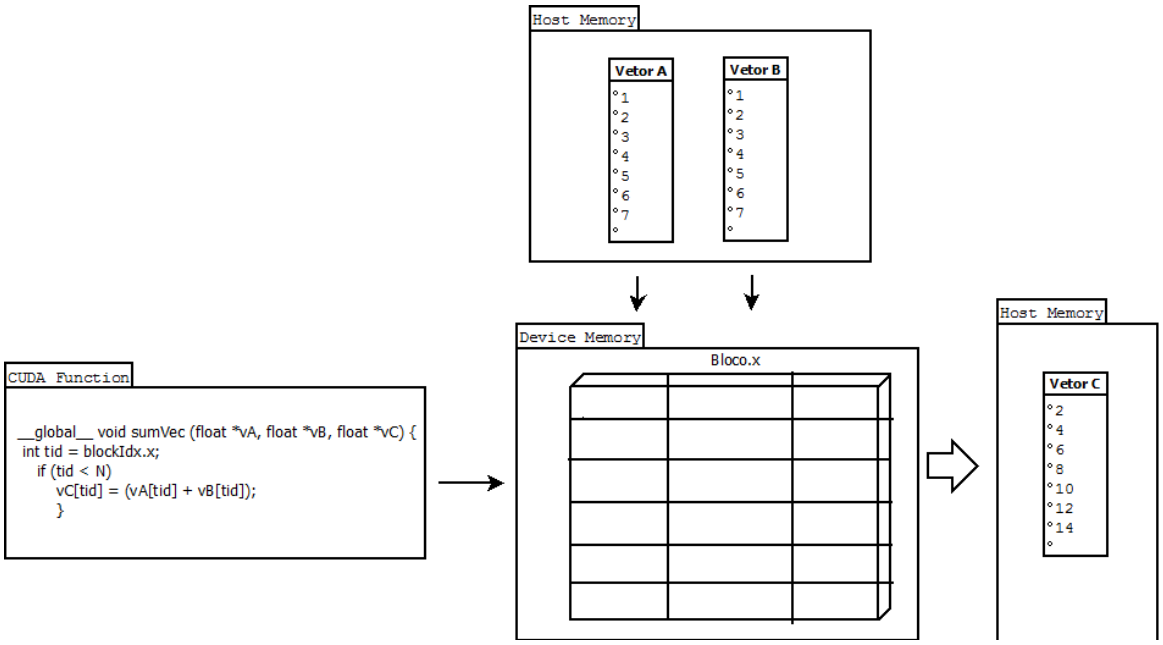

Figura 12 - Soma de vetores paralelo.

Neste caso dizemos então que vC[tid] = (vA[tid] + vB[tid]), onde "tid" indexa o endereço da *thread* no bloco, onde foi armazenado na tupla para a execução da operação, o que seria equivalente ao vC[i] = vA[i] + vB[i], com a grande diferença de que "tid" não é usado como passo para a execução em cada tupla dos vetores, pois a execução da operação será feita de uma só vez em todo o bloco de *thread*.

Após a execução da operação, o vetor de resultados é movido para a memória principal (*Host Memory*). Baseado nessa arquitetura, o programa CUDA realiza parte do modelo estatístico usando blocos de *threads* na GPU, para realizar a carga, processamento dos dados e posterior armazenamento desses dados, nesse caso as séries históricas com dados curados pela aplicação do modelo estatístico.

3.1.4 *Workflow* de processamento sequencial

Neste trabalho, também foi desenvolvido um *workflow* concreto para processamento do tipo sequencial, sendo que a única diferença em relação ao paralelo é a ausência de utilização da integração dos *workflows* com programas em CUDA, nesse caso substituídos por módulos escritos em *Python*, utilizando a ferramenta "*python source*" do próprio *VisTrails*. Sua implementação se baseou no mesmo aspecto conceitual representado pelo *workflow* abstrato.

Isso significa que ambas versões são comapatíves, servindo para comparações entre as execuções realizada entre cada uma das versões (paralela e sequencial)

3.2 Banco de Dados Meteorológicos Curados

Ao término do processamento dos *workflows*, os dados brutos, curados e a sua proveniência são armazenados em repositórios de dados relacionais. Neste caso, o modelo de dados proposto é uma adaptação do modelo concebido por (FILHO et al., 2013). No novo modelo, tratamos cada execução (*run*) do *workflow* concreto como um experimento em

separado. Em seguida, armazenamos os dados das estações utilizadas em cada experimento, assim como a parametrização definida pelo pesquisador, os dados brutos de cada estação conforme extraídos do sistema HidroWeb da ANA também são armazenados, e em seguida a execução de cada workflow, estes têm seus dados curados de saída armazenados para que sejam utilizados posteriormente pelos pesquisadores e suas equipes.

3.2.1 Modelo de Dados

O modelo da dados relacional utilizado nesta dissertação é uma adaptação do modelo desenvolvido por (FILHO et al., 2013). Reutilizamos o modelo pois já existem diversas pesquisas e sistemas *Web* no grupo ao qual esta dissertação está vinculada. O modelo foi adaptado para tratamos cada rodada (*run*) do *workflow*, como um experimento *in silico*, por isso a necessidade de identificação individual do mesmo. Além disso, é necessário manter os dados das estações utilizadas no experimento, assim como a parametrização definida pelo pesquisador para a execução dos *workflows*. Os dados brutos coletados de cada estação são extraídos do HidroWeb ANA também são armazenados e, em seguida, à execução de cada *workflow* também são armazenados os dados curados. Esses têm seus dados (de saída) que são armazenados juntamente com a proveniência do experimento.

A seguir encontra-se a descrição das tabelas utilizadas para o armazenamento dos dados curados e os dados relativos a proveniência prospectiva.

A Tabela Experimento, contém os dados relativos aos experimentos executados, lembrando que cada rodada (*run*) do *workflow* é tratada como um experimento, além disso estão incluídos os dados da parametrização utilizada da rodada. Essas informações são importantes para a reprodução de resultados.

A Tabela Pesquisadores, possui os dados cadastrais de cada um dos usuários do *workflow*, que os identifica como pesquisadores.

A tabela Estações_Ana comportam os dados das estações meteorológicas, incluindo tipo, responsável mantenedor, e a localização geográfica (altitude e longitude).

O conjunto de Tabelas, estados, município, bacias e sub bacias possuem informações que permitem detalhar as informações sobre a geografia local da área coberta pela estação, informações como bacia e a sub bacia. São de importância para a criação de relatórios (possibilidade futura), e para a definição de parâmetros do experimento a ser realizado.

A Tabela DadosBrutos recebe os dados gerais do arquivo de séries históricas de uma estação. Os dados de leitura de chuvas ficam na tabela ChuvasBruto, havendo uma relação entre as tabelas por meio da chave idDadosBrutos, que liga o valor de determinado dia de chuva em determinado mês, ao arquivo e naturalmente a estação que essa leitura se refere.

A tabela Regressão recebe os dados curados, ou seja, recebe os dados das médias mensais de chuvas, após o tratamento das falhas por meio do modelo estatístico.

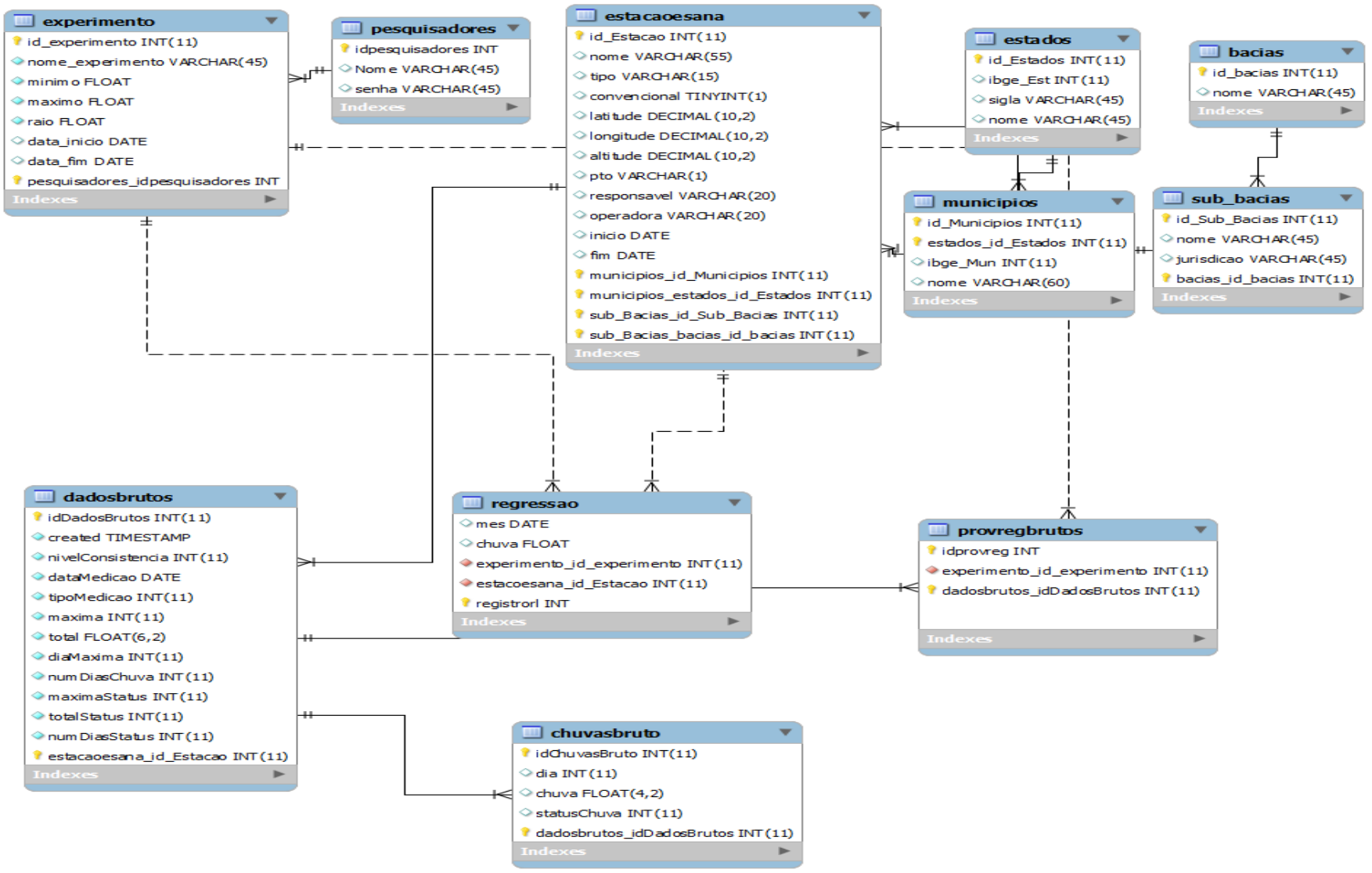

Figura 13 - Modelo de dados

A tabela *Provregbrutos* faz a ligação entre as tabelas anteriores a fim de estabelecer a ligação entre as tabelas apresentadas anteriormente e promover a coleta dos dados de proveniência, conforme maior detalhamento na seção seguinte.

3.3 Arquiterura *Metflow*

Nesta pesquisa de dissertação, batiza-se a arquitetura desenvolvida como *Metflow*, nome que visa representar a junção de *workflow* e Meteorologia.

A Figura 14, representa a arquitetura *Metflow*, sua metodologia (uso de *workflows*), tecnologias e modelos usados (Local e Distribído).

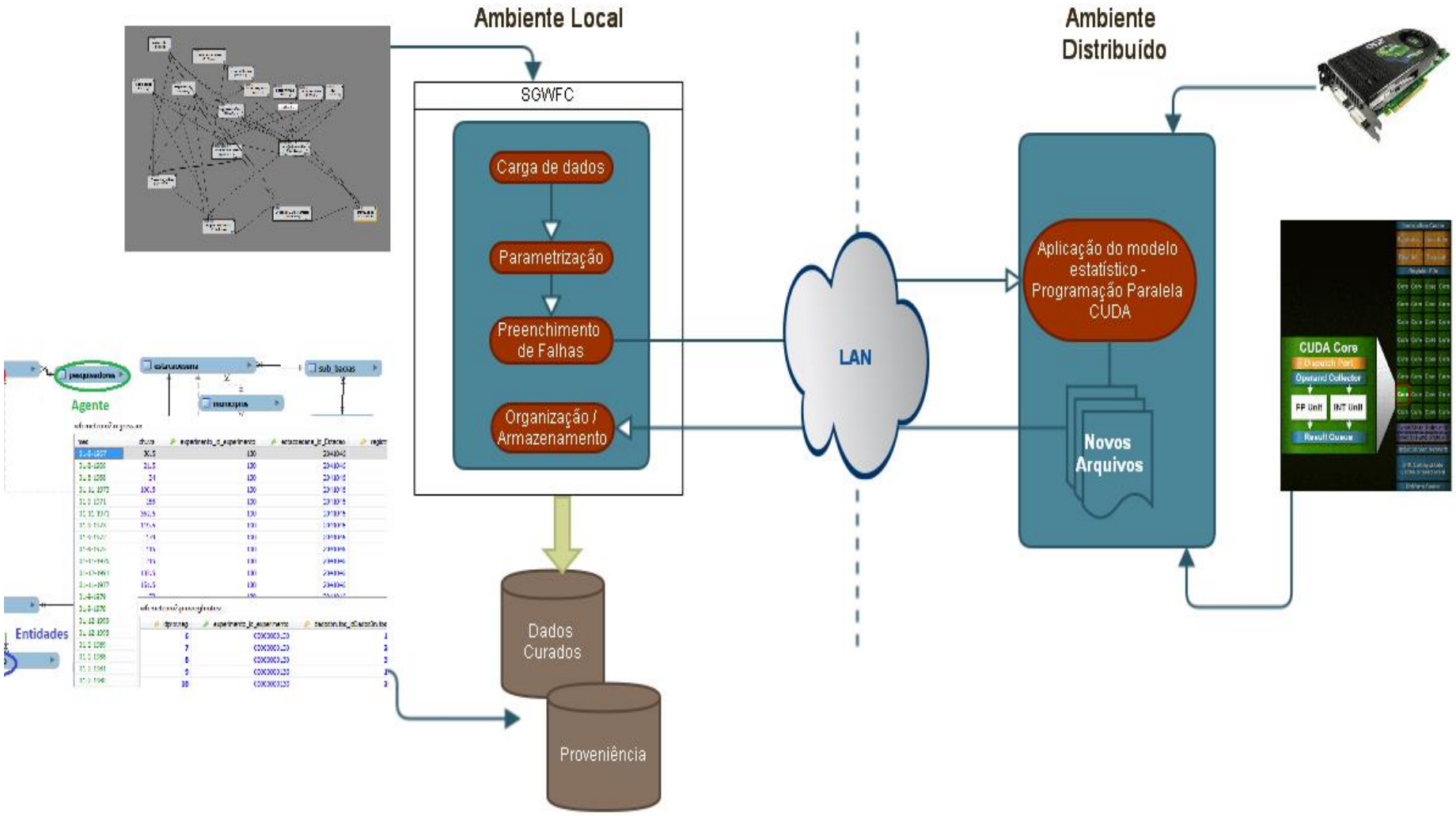

Figura 14 Arquitetura *metflow* conceitual

Tomando como referência a nuvem, que especificamente visa representar uma LAN (Local Área *Network*), por retratar o ambiente deste trabalho, a esquerda como item principal o *workflow* (*Metflow*) , contido em seu SGWFC, ligando-se a base de dados curados, que deve ser onde o *workflow* deve depositar o resultado de seu processamento (principalmente o resultado da aplicação do modelo estatístico), ligando-se a esta base por sua vez, está o modelo conceitual do banco de dados, este itens estão associados ao ambiente local de execução.

A esquerda da nuvem, a aplicação desenvolvida em CUDA, e os arquivos que são resultados da execução desta aplicação, a figura da placa de vídeo e o diagrama de organização de GPU, ilustram que o ambiente distriuido onde será executada a aplicação conta com esse recurso.

Na figura 15, apresenta-se a sequencia de manipulação dos dados proposta na arquitetura *Metflow*.

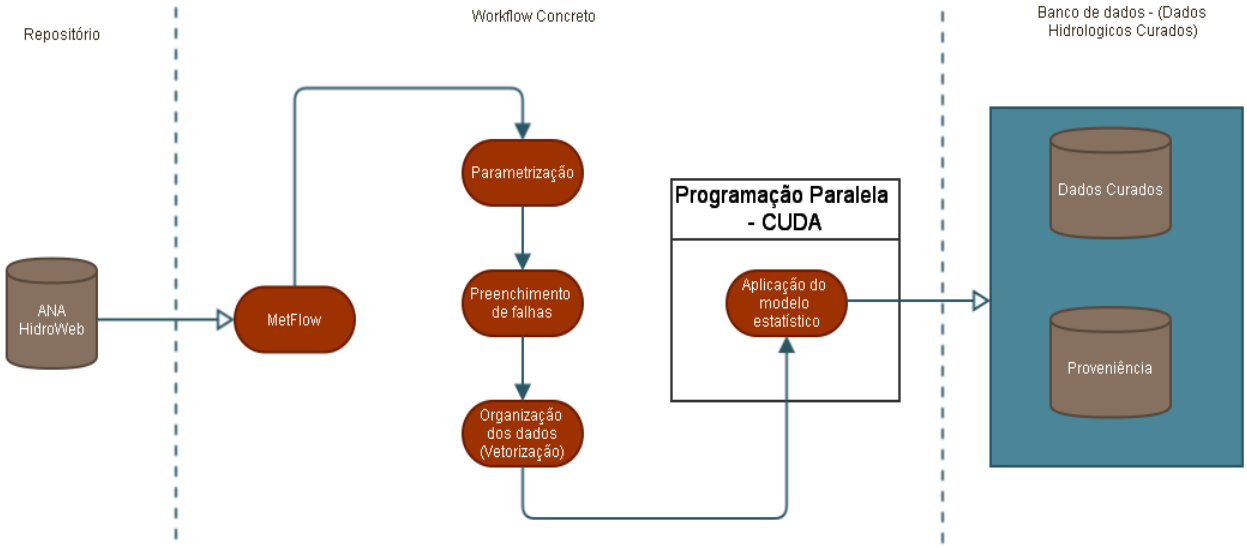

Figura 15 *Metflow*, fuxo de dados

Começando a esquerda temos a representação do repositório da dados da ANA, conhecido como Hidroweb, a partir da seleção das estações que participarão do experimento, estes são submetidos ao *Metflow*, que por sua vez, permite a parametrização (pesquisasor), realiza o preenchimento de falhas (etapa automática, realizada pelo *Metflow*), organização e envio do dados a serem submetidos pela aplicação que contém o modelo estatístico (CUDA) e por fim o armazenamento dos dados envolvido no experimento da base de dados.

Agora na Figura 16, apresenta-se o confronto entre o proposto para arquitetura *Metflow* e os artefatos desenvolvidos (concretos), baseados na arquitetura proposta.

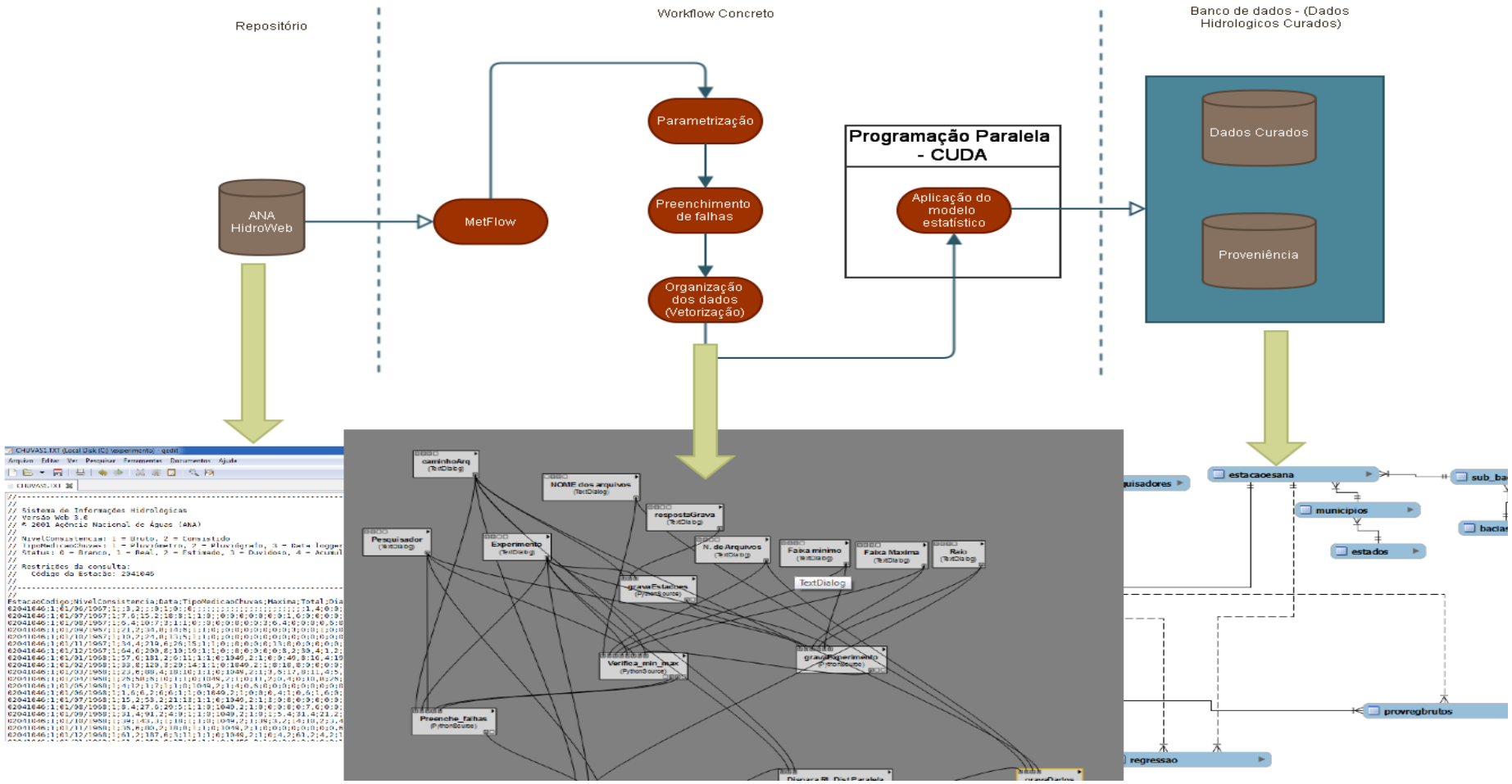

Figura 16 *Metflow* conceitual X concreto

Da esquerda para direita temos, a representação o repositório de dados, associado a um arquivo de dados de série meteorológica extraído do Hidroweb, ao centro a arquitetura *Metflow*, associada ao *workflow* concreto desenvolvido, e a direita a representação do modelo de dados da arquitetura *Metflow*, associada ao modelo de banco de dados desenvolvido.

As Figuras 14, 15 e 16, visam demonstrar a ligação entre a proposta, portotico e artefatos de software desenvolvidos durante esta pesquisa, naturalmente que visa-se também a exposição de alguma das etapas de pesquisa deste trabalho de dissertação.

4. RESULTADOS EXPERIMENTAIS E DISCUSSÃO

Conforme apresentado nos capítulos anteriores, além do desenvolvimento dos *workflows* abstrato e concreto e seus módulos de computação paralela, utilizando programação em CUDA, este capítulo mostra os produtos de *software* desenvolvidos e os experimentos realizados neste dissertação, também serão descritos os resultados de desempenho da execução dos *workflows* apoiados nos códigos de programação paralela.

4.1.1 Ambiente de Desenvolvimento

O desenvolvimento do produtos de *software* deste trabalho é iniciado, com a construção do *workflow* concreto, conforme proposta apresentada anteriormente. Para isso é utilizado o SGWC Vistrails na versão 2.1 beta, que foi selecionado por obter maior pontuação na matriz de decisão apresentada anteriormente na seção 2 na Tabela 1. Os módulos de processamento paralelo tiveram seus códigos gerados em linguagem CUDA, utilizando compilador NVCC da Nvidia (versão 5.5.0). Esse compilador foi escolhido por ser o padrão da linguagem, e por ter compatibilidade sem adaptações, aos módulos do *workflow* concreto. Foi utilizado o banco de dados relacional *MySQL* na versão 5.6, para gravação da proveniência retrospectiva, dos dados meteorológicos brutos e curados. Esse banco foi selecionado por se adequar as condições do trabalho e por já estar em uso nos projetos do Grupo de Pesquisa Meteoro da UFRRJ, o que permite maior poder de transporte e compatibilidade entre os dados que envolvem esse grupo..

A tabela 3 indica os softwares e suas versões utilizados nos experimentos desta dissertação. Ela descreve os recursos para a versão do *workflow* científico executado em ambiente local. Na tabela 4 temos as configurações de *hardware* utilizada para realização dos experimentos.

Tabela 4 - Ambiente desenvolvimento SOFTWARE

Software	Tipo	Versão	Arquitetura (32 ou 64 bits)
Vistralis	SGWC	2.1	32 bits
NVCC (CUDA)	Compilador	5.5.0	32 bits
MySQL	SGBT	5.6.11	64 bits
Windows 7	S.O	H.P	64 bits

Tabela 5- Ambiente desenvolvimento HARDWARE

Processador	Memória Ram	Pl. Vídeo
Intel Core i5-520M	SO-DIMM | DDR3 SDRAM	NVIDIA GeForce GT 335M
2.933 GHz	DDR3-1066 (533 MHz)	1080 MHz
Cores = 2 | Threads = 4	2 x 2 Gb	1 GB
		CUDA Cores = 72

4.1.2 Desenvolvimento do *workflow* concreto e coleta da proveniência prospectiva e versionamento de dados

Assim como no processo de desenvolvimento de qualquer tipo de *software*, o desenvolvimento de *workflows* também possuem etapas. No entanto com a proposta de coleta da proveniência prospectiva, temos a vantagem de evoluir no desenvolvimento, sem perder a possibilidade de executar e analisar as versões anteriores de um *workflow*, sem a necessidade de um software de controle de versão ou a criação de ambientes distintos para esse tipo de execução ou teste.

Nas Figuras 17 e 18, estão exibidos uma versão *workflow* e o formulário de controle de versões (*history*) do *Vistrails*, conforme proposta deste trabalho, essas figuras demonstram a forma de coleta e o funcionamento da proveniência prospectiva.

Em uma de suas versões finais o *workflow* apresenta o seguinte formato.

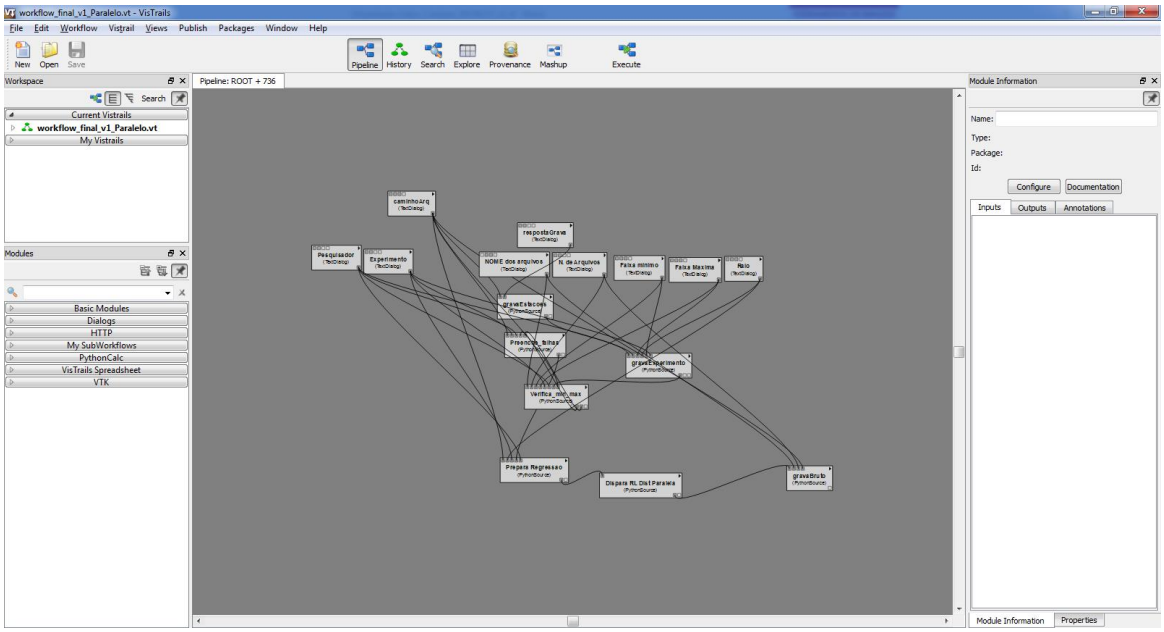

Figura 17 - Área de trabalho *Vistrails* (SGWFC) e versão do *workflow* concreto.

No entanto, se aplicarmos a função *history*, disponível na barra de ferramentas do *VisTrails*, pode-se observar os eventos ocorridos nas etapas de desenvolvimento. Nesse caso o *VisTrails* exibe um diagrama com os eventos que ocorreram no processo de desenvolvimento, permitindo ao pesquisador analisar códigos, e também executar o *workflow* especificamente na etapa selecionada. Este recurso é demonstrado na Figura 18.

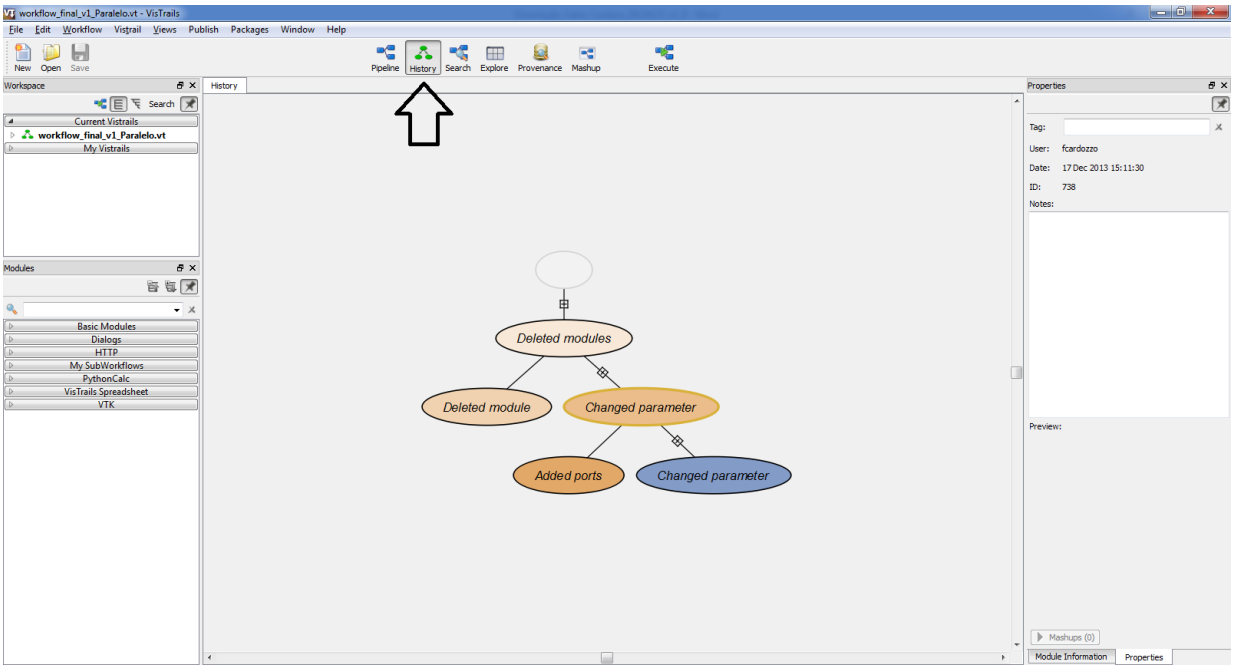

Figura 18 - Proveniência prospectiva (função *history)* nativa do *Vistrails.*

Cada uma das elipses da figura 18, representam um tipo de evento ocorrido no desenvolvimento (alteração, inclusão ou edição) do *workflow*. Quando selecionado um desses eventos o *Vistrails* apresenta a versão do *workflow* antes da ocorrência do evento, conforme apresenta Figura 19.

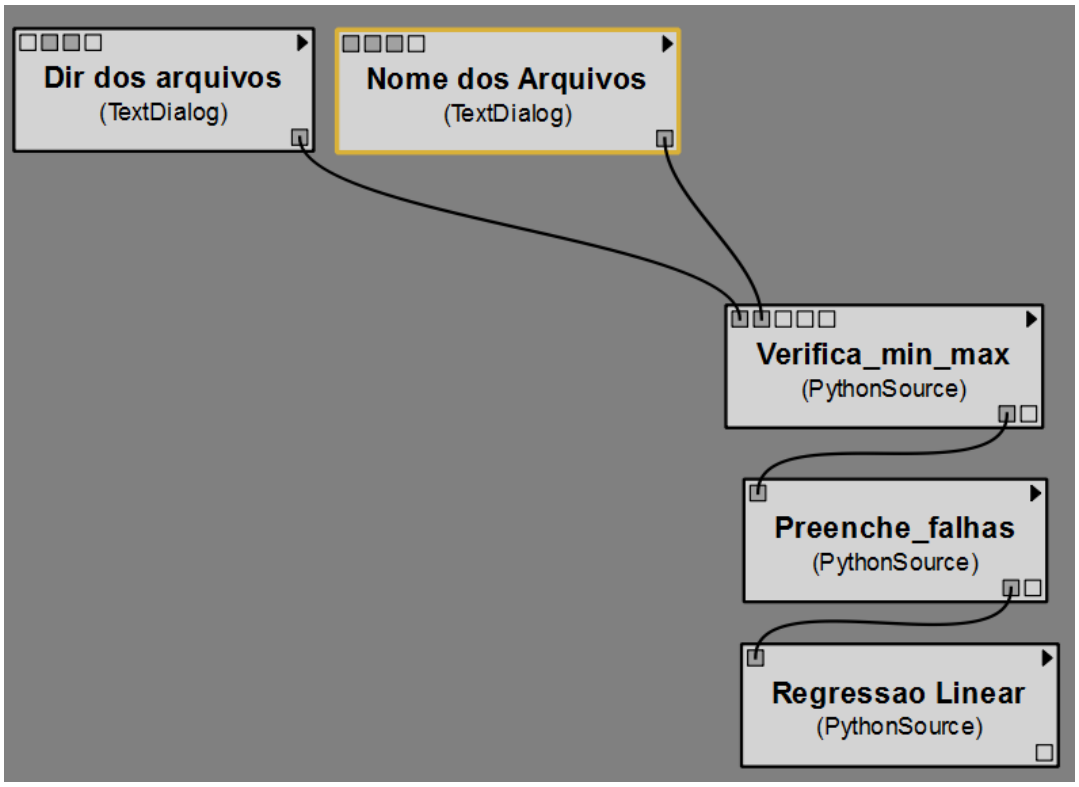

Figura 19 - *Workflow* concreto, versão anterior

A partir deste ponto, foi dedicado o espaço para descrição dos módulos criados para o *Workflow* e suas funções, essa descrição é iniciada pela tabela de descrição dos módulos (tabela 8), seguida pela imagem que ilustra a versão final do *Workflow* (figura 20).

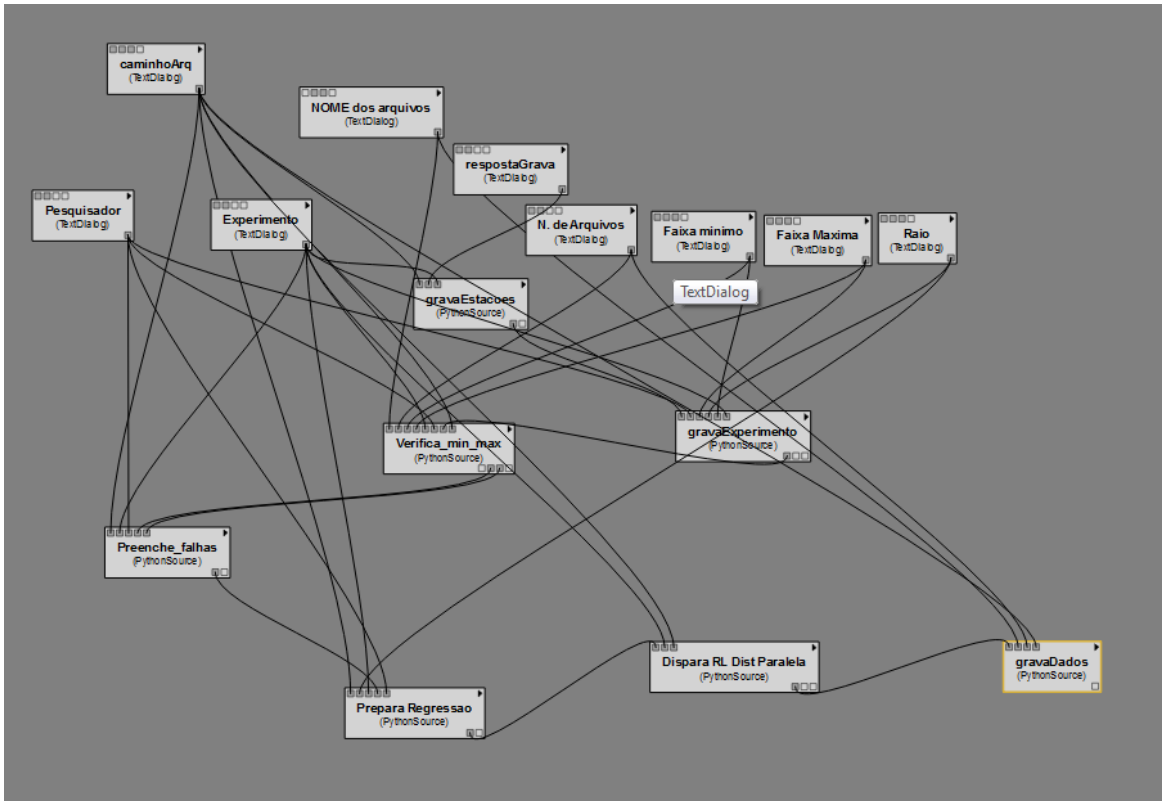

Figura 20 - *Workflow* concreto versão final

A Tabela 6, descreve as ações envolvidas na execução do *workflow* concreto desenvolvido nesta dissertação para efetuar o tratamento de dados e preenchimento de falhas.

Tabela 6 - Tabela de descrição de ações do *worflow* concreto

Ação	Descrição	Tipo
Definir de diretório	Recebe como parâmetro os nome dos diretórios que contém os arquivos texto com as séries hidrológicas recuperados do sistema Hidroweb	Sequencial
Definir de nome dos Arquvios	Armazena o nome único utilizado nos arquivos ANA, Ex: CHUVAS*nn*.	Sequencial
Definir número de Arquivos	Quantidade de arquivos, que serão processados no experimento.	Paralelo
Definir nome do experimento	Armazena o nome do experimento	Sequencial
Pesquisado	Armazena o nome do pesquisado.	Sequencial
Parametrizar faixa mínimo	Parâmetro que define o valor mínimo de leitura de chuva, aceito como válido no experimento em questão	Sequencial

Tabela 6. Continuação

Parametrizar faixa máximo	Parâmetro que define o valor máximo de leitura de chuva, aceito como válido no experimento em questão	Sequencial
Parametrizar raio	Faz a leitura do valor que define, o raio máximo, para considerar ou não as leituras de uma estação.	Sequencial
Verificar mínimos e máximos	Avalia se uma leitura está ou não dentro das condições parametrizadas.	Sequencial
Preencher falhas	Substitui valores em branco (falhas), ou valores fora da parametrização por -9999.99.	Sequencial
Preparar dados da regressão	Cria os vetores de dados para aplicação do modelo estatístico	Sequencial
Executar regressão paralela	Executa o módulo CUDA de RL, localmente ou em servidor remoto.	Paralelo
Gravar estações	Grava dados das estações ANA envolvidas no experimento	Sequencial
Gravar dados	Faz a leitura dos arquivos de dados brutos e curados, envolvidos no experimento e os grava em banco de dados..	Sequencial

4.2 Experimentos Realizados

A ANA possui 4.543 das estações hidrometereológicas brasileiras, de um total de 14.822 no território nacional.

Os experimentos realizados, utilizaram a base de dados histórica do sistema Hidroweb da ANA, arquivos hidrológicos, contendo dados sobre precipitação pluvial, as estações selecionadas que estão dentre as 555, que cobrem o estado do Rio de Janeiro, sendo o principal critério de seleção das estações, o tamanho e ano de início e fim, utilizando séries com tamanho igual ou superior a 20 anos a partir de 1960, sendo esse critério atendido por 77 estações.

Cada um dos experimentos levou em conta o número de estações utilizadas, os registros contidos, os registros com falha, os registros corrigidos e o tempo de execução, os parâmetros de execução serão citados e usados da mesma forma tanto na execução sequencial quanto a execução no ambiente paralelo.

4.2.1 Teste de Desempenho

Foram realizados testes de comparação de tempo de execução de cada experimento utilizando o *workflow* sem o uso do código CUDA, neste caso trata-se de um *workflow* sequencial, e a execução do *workflow* que utiliza o código CUDA, neste caso *workflow* paralelo.

Cada experimento usou os mesmos parâmetros para ambos *workflows* e cada rodada conta com um volume maior de dados que é função do aumento do número de estações envolvidas em cada rodada do *workflow*. Cada experimento foi executado cinco vezes e os valores representados nos gráficos indicam os valores médios de tempo de execução.

Os teste estão divididos em execuções que usam 17, 36 e 77 estações, essa divisão é feita com o intuito de acompanhar o desempenho, de acordo com o crescimento do volume de dados, não existindo regra matemática.

O gráfico da Figura 21 é o comparativo entre os tempos de execução dos *workflow* sequencial e o paralelo, em um experimento utilizando 17 estações, com aproximadamente 8838 registros sendo processados.

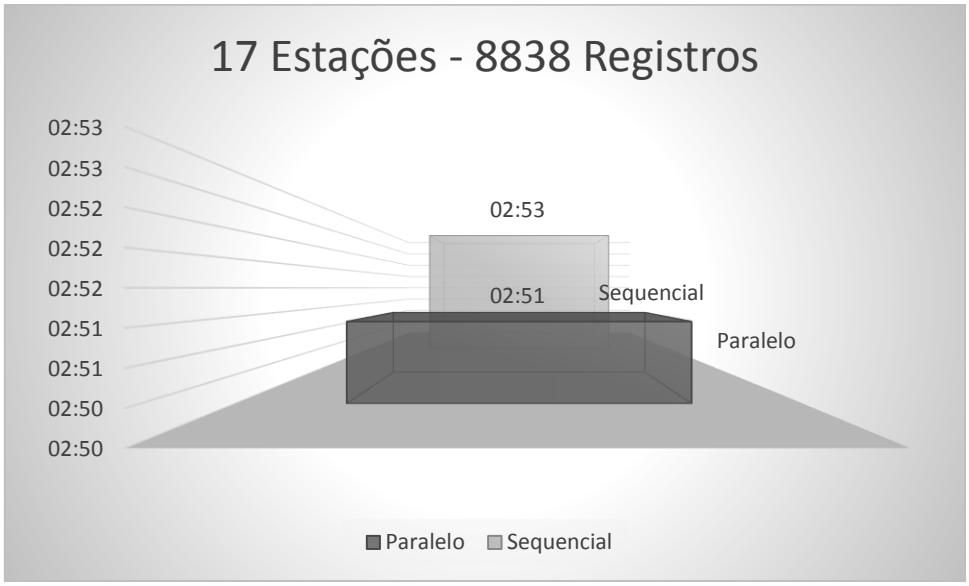

Figura 21– Comparação do tratamento de dados de 17 estações, *workflow* paralelo e sequencial, tempo em minutos

Ressalta-se que a diferença no tempo de execução entre as duas configurações é de apenas 2 segundos. A seguir temos um novo experimento utilizando 36 das 77 estações selecionadas e diferença no tempo de processamento se torna maior, conforme ilustrado no gráfico da Figura 22.

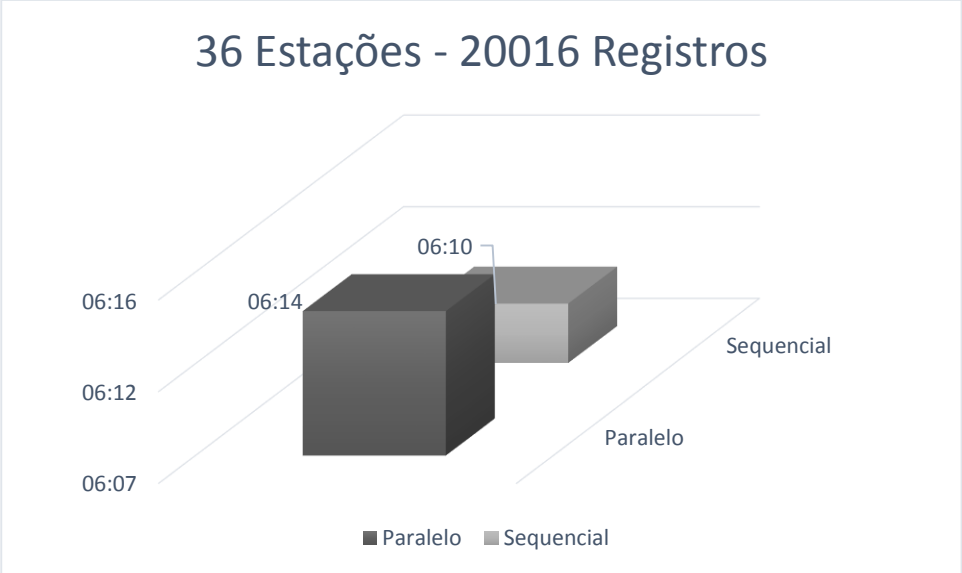

Figura 22 - Comparação do tratamento de dados de 36 estações, *workflow* paralelo e sequencial

Na primeira bateria de testes, utilizando 36 estações, uma surpresa quanto ao resultado no tempo de execução, o *workflow* sequencial superou a execução do *workflow* paralelo, o que motivou uma reavalição do código paralelo. Após uma revisão na etapa de carga dos vetores (no modulo de regressão paralela), o resultado torna a ser novamente melhor para código paralelo. Este tipo de comportamento não era esperado, demonstrando a necessidade de avaliação rotineira de códigos paralelos e suas tecnologias.

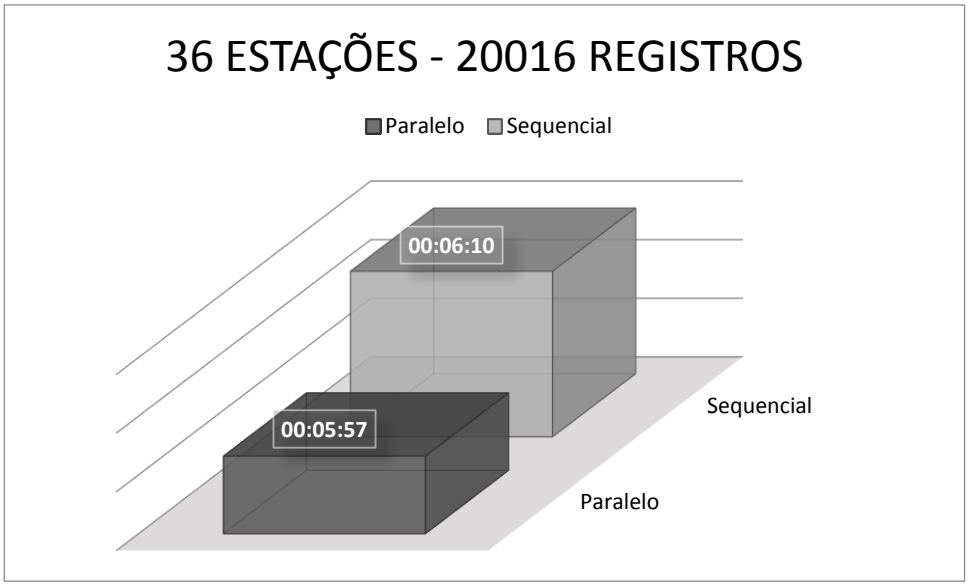

Figura -23 Comparação do tratamento de dados de 36 estações, *workflow* paralelo e sequencial, tempo em minutos

Após a revisão no código paralelo, em um teste com carga superior demonstra, que quando observado o desenvolvimento do código assim como de seus resultados são importantes. Neste teste (figura 23) o *workflow* paralelo apresenta melhor desempenho em comparação ao sequencial.

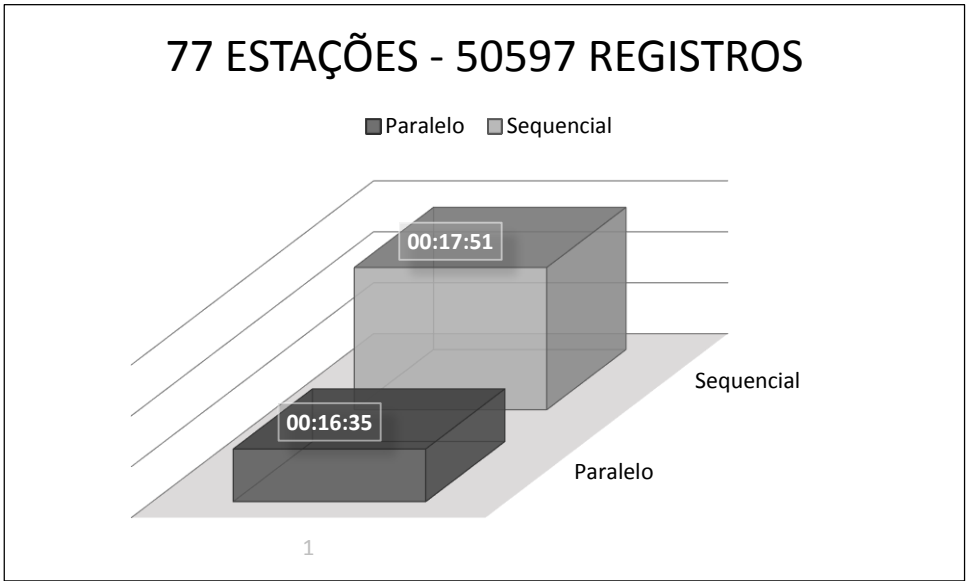

Figura 24 - Comparação do tratamento de dados de 77 estações, *workflow* paralelo e sequencial, tempo em minutos

Na figura 24, agora utilizando 77 estações, percebe-se o crescimento da vantagem do *workflow* paralelo, sendo esse o comportamento esperado neste teste.

4.2.2 - Teste de desempenho ii – threads.

Após os testes de comparação entre o *workflow* sequencial e paralelo, realizou-se testes de desempenho Paralelo Vs Paralelo, onde o paramêtro de comparação é o aumento do número de *threads* por bloco, em cada uma das execuções.

No gráfico da Figura 25, compara-se a execução do *workflow* paralelo em um experimento de 77 estações, variando o número de *threads* por bloco.

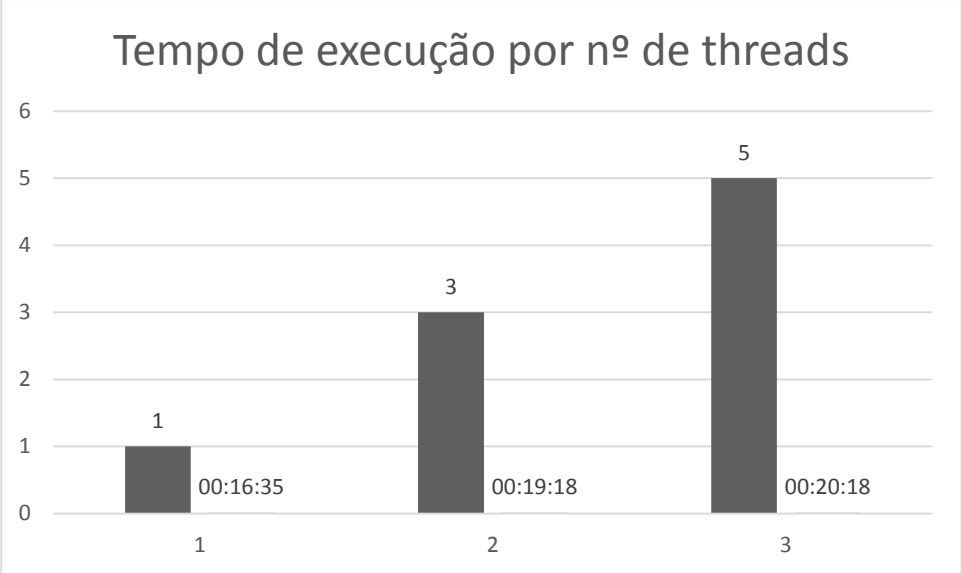

Figura 25 - Comparação dos tempos de execução, com mais de uma *thread*, tempo em minutos

Neste experimento, pode-se observar, que a solução em questão possui melhor desempenho quando executando somente 1 *thread* por bloco, dado que o aumento do tempo de execução, a cada acréscimo de *threads*, implica no aumento percentual no tempo de processamento, conforme apresentado no gráfico da Figura 26.

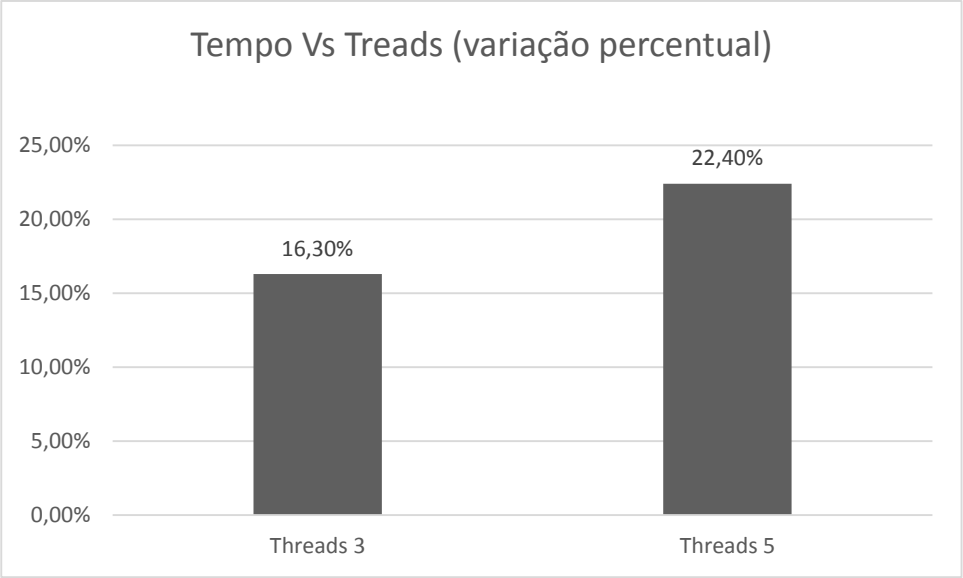

Figura 26 - Aumento percentual do tempo de execução, utilizando 3 e 5 threads

Com o aumento no tempo de execução do experimento nos casos de *threads* por bloco **> 1**, pode-se dizer que para o problema o deste trabalho, o custo computacional para criação e gerenciamento de um número maior de *threads*, não se mostra eficiente.

4.2.2 Testes de qualidade de dados

Além do desempenho, outro fator analisado foi o percentual de correção de dados alcançado por rodada de experimento. Neste caso, avaliamos diretamente o número de registros (em meses) que possuíam falhas, e o número de registros que puderam ser inseridos a partir da execução do *workflow*.

Os gráficos das Figuras 27 e 28, mostram o número de meses com falhas e o número de recuperações.

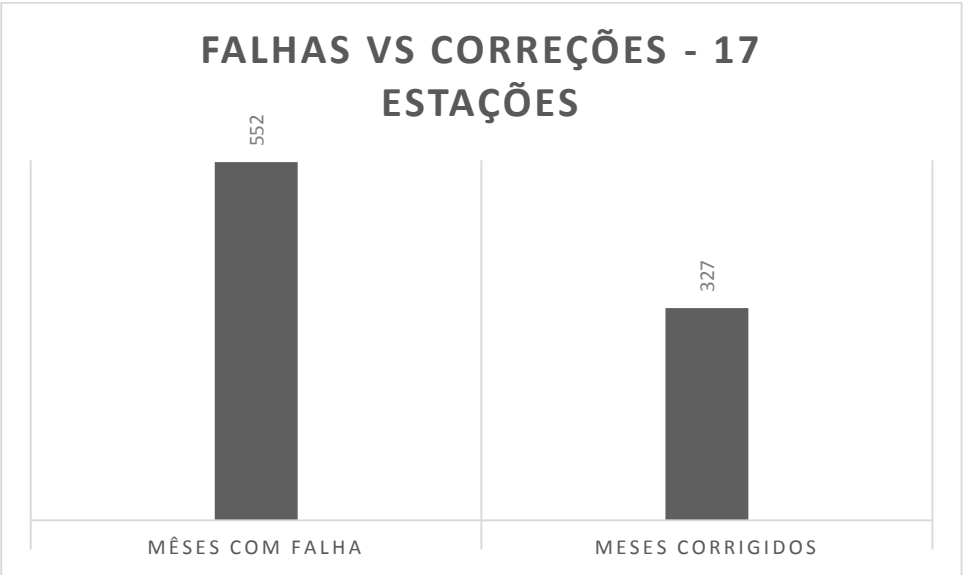

Figura 27 - Relação entre meses com falhas e meses corrigidos após a execução do *workflow* – 17 Estações

No gráfico da Figura 27 observou-se que o número de registros com falhas, que é de 552 e o número de correções, que é de 327, percentualmente a taxa de correção fica em torno de 52,2%, neste experimento foram utilizados os dados de somente 17 estações.

Essa diferença de 225 registros ocorre pois, dependendo do número de estações envolvidas e da parametrização, nem sempre é possível obter dados de estações doadoras, que permitam a aplicação da correção de falhas.

A seguir realizou-se mais experimentos com maior volume de dados. O gráfico da Figura 28, ilustra um número maior de estações. Essas comparações visam saber se o aumento do número das estações envolvidas, melhora o percentual de correção de falhas.

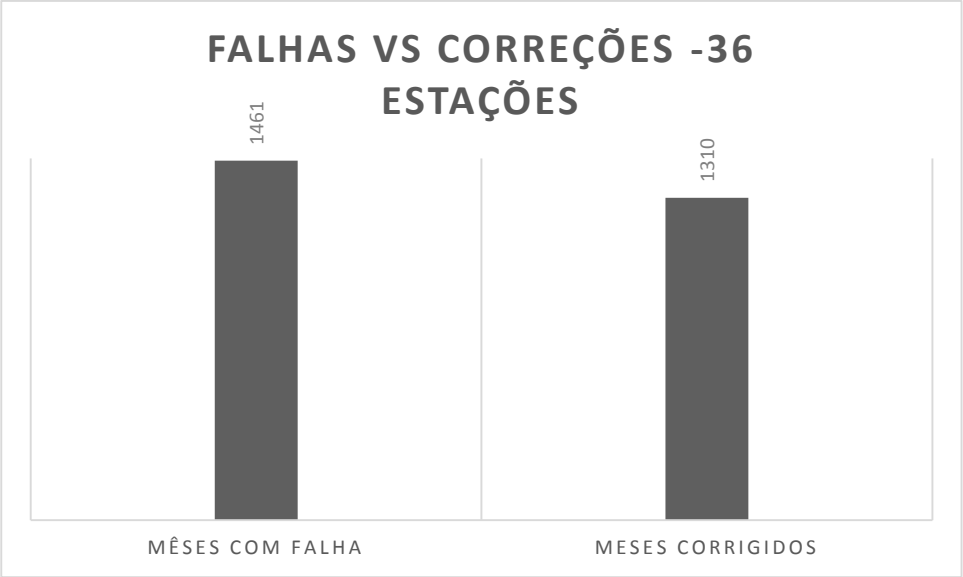

Figura 28 - Relação entre meses com falhas e meses corrigidos após a execução do *workflow* – 36 Estações

No gráfico da Figura 28 pode-se observar, que o aumento das estações envolvidas aumenta consideravelmente o percentual de correções, indo dos 59,2% usando 17 estações do primeiro caso, para 89,6% usando 36 estações. Com isso temos o comportamento esperado do modelo estatístico, onde previámos que um maior número de estações (dados) permitiria um melhor aproveitamento do método estatístico.

Ainda realizou-se testes com 89 estações com o objetivo de confirmar se os resultados são melhores quando aumentado o número de estações (dados).

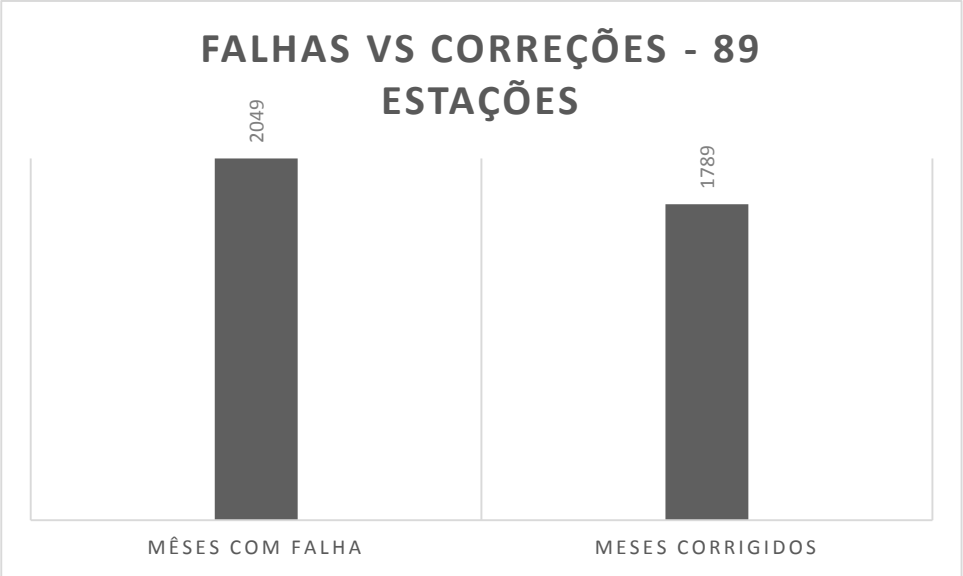

Figura 29 - Relação entre meses com falhas e meses corrigidos após a execução do *workflow* – 89 Estações

Nesse terceiro teste, utilizando 89 estações, neste caso obtevi-se um índice de 87,7%, contra os 89,6% das 36 estações. No entanto, este resultado confirma o comportamento esperado, pois na utilização das 89 estações, utilizou-se as séries praticamente todos os arquivos de estações do Rio de Janeiro, incluindo os arquivos alto número de falhas, e de estações mais novas com séries de dados menores.

4.4 Abordagem da Coleta De Dados De Proveniência Retrospectiva

Conforme a proposta apresentada, este trabalho visa contribuir para a solução do problema da correção das séries históricas de dados hidrológicos, com *workflows* baseados em proveniência de dados. Para isso realizamos a coleta de dados proveniência retrospectiva, conforme orientação do PROV-DM, modelo citado da fundamentação teórica desse trabalho.

A partir do modelo PROV-DM, buscou-se a relação entre, Atividade, Entidade e Pessoa envolvida (Agente). Na análise do problema, é sugerida a seguinte relação, Dado Bruto (entidade), usada pelo Experimento (Atividade), deriva Dado Curado (Entidade), associado ao Pesquisador (Agente), pelo Experiemnto (Atividade). Os dados brutos são usados por experimentos (*run do workflow*),que geram dados curados, o experimento por sua vez está associado ao pesquisador.

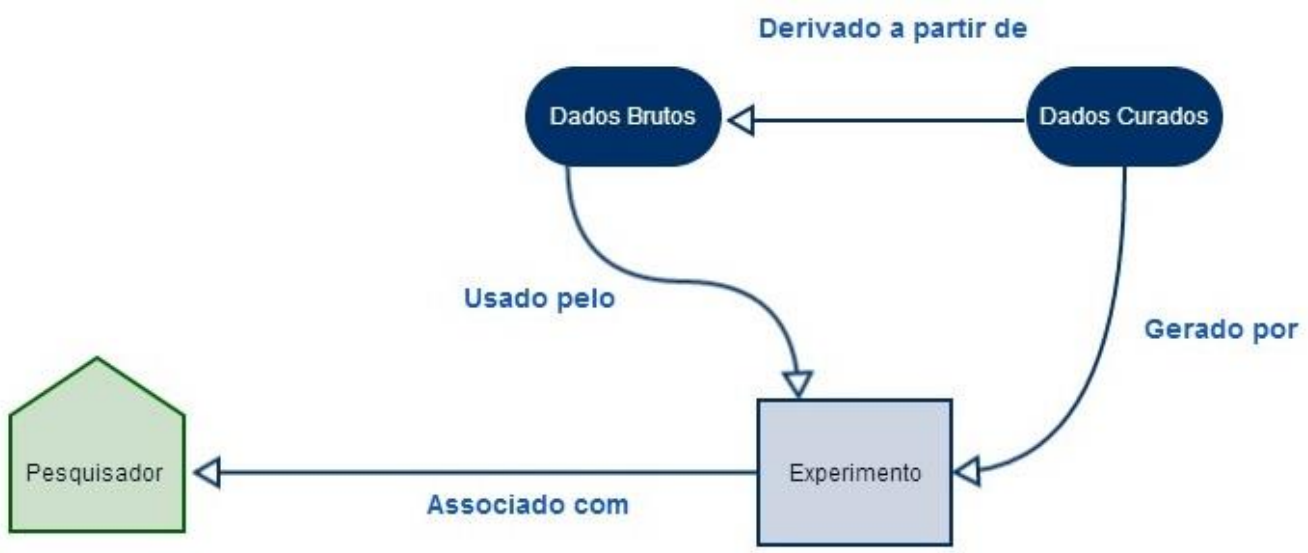

Figura 30 - Aplicação do modelo Prov-DM ao modelo de dados do estudo

Neste caso o registro de dados permite conhecer quais dados brutos (Entidade) são utilizados em um determinando experimento (Atividade), o que naturalmente permite identificar o pesquisador (Agente), que inicia o processo.

Além da relação do parágrafo anterior tem-se ainda as relações dos dados gerados pelo *Workflow*, nesse caso os dados que são resultado da aplicação do modelo estatístico (regressão linear). Essas relações atendem ao PROV-DM, e são importantes na visualização do grupo de dados resultantes e a influência do conjunto de parâmetros do experimento e seu respectivo pesquisador.

Na prática, os coletores desses dados, são *scripts* de códigos escritos em *Python*, executados pelo *workflow* após cada rodada (*run*), esses *scripts* fazem a leitura dos arquivos de dados brutos utilizados no experimento, e os arquivos de dados curados (novos). Após a leitura esses dados são inseridos nas respectivas tabelas a que pertencem.

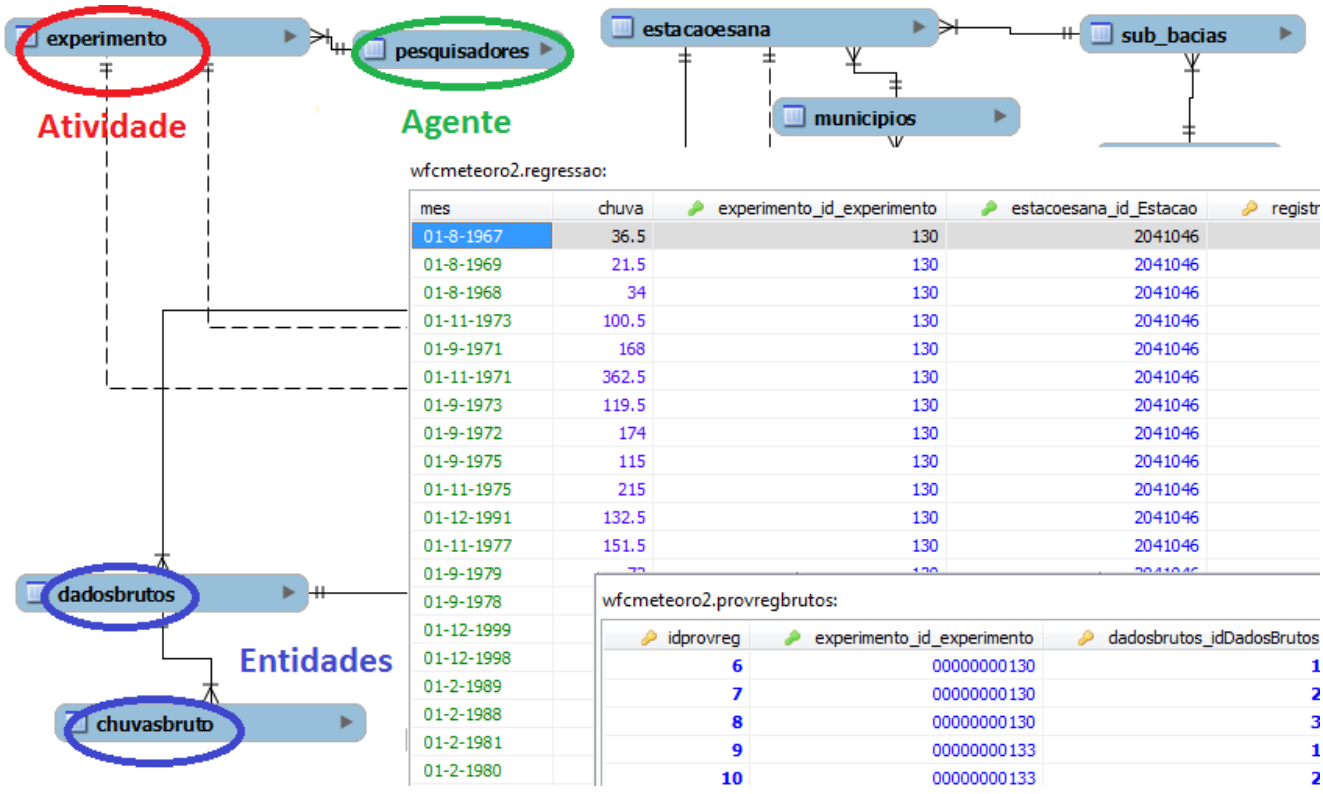

Figura 31 - Apresentação de dados de proveniência, baseado no modelo e apresentação de dados

5 CONCLUSÃO

Neste capítulo são apresentas as contribuições deste trabalho, além da exposição do caminho percorrido até sua finalização.

5.1 Introdução

Conforme relatado, este trabalho envolveu o desenvolvimento de técnicas ainda pouco exploradas, como o desenvolvimento de *workflows* para ambientes GPU e a combinação desses com a computação paralela para manipulação de grandes volumes de dados meteorológicos. Durante o processo de pesquisa apresentamos pelo menos três trabalhos em eventos locais, que são *"WORKFLOWS* CIENTÍFICOS DISTRIBUÍDOS EM AMBIENTE GPU PARA MANIPULAÇÃO DE DADOS METEOROLÓGICOS" apresentado no VIII Fórum de Pós-Graduação da UFRRJ em 2013, e ainda "GERÊNCIA DE TRIPLAS RDF DE PROVENIÊNCIA EM EXPERIMENTOS CIENTÍFICOS EM LARGA ESCALA", apresentado na 1ª. Reunião Anual de Iniciação Científica da UFRRJ - I RIAC. 2013. Durante este trabalho de dissertação possível realizar o trabalho em conjunto com quatro alunos de Iniciação Científica do curso de Sistemas de Informação da UFRRJ, a saber: Geraldo Lemos Filho, Mário Junior Ribeiro da Costa, Thiago Barbos e mais recentemente Luan Soares Andrade que apresentou o trabalho "DESENVOLVIMENTO DE *WORKFLOWS* CIENTÍFICOS PARA AUXILIAR A GERÊNCIA DE DADOS METEOROLÓGICOS NO ESTADO DO RIO DE JANEIRO" na 2ª. Reunião Anual de Iniciação Científica da UFRRJ - II RIAC. 2014.

Outro evento importante do qual participamos perfom foi o minicurso de "APLICAÇÕES BIOMÉDICAS EM PLATAFORMA COMPUTACIOANAIS DE ALTO DESEMPENHO EM PLACAS GPU´S", realizado no LNCC (Laboratório Nacional de Computação Científica), em parceria com a Universidade de Málaga – Espanha. Este curso foi realizado entre 21 de Outubro e 01 de Novembro de 2013, e envolveu assuntos relacionados a arquiteturas de placas GPU com suporte a CUDA e desenvolvimento de aplicações usando computação paralela baseada em GPU. O curso foi de fundamental importância para a execução desta dissertação.

5.2 Contribuições do Estudo

Como principal contribuição temos os *workflows* concretos, que são artefatos capazes de realizar o trabalho de pré-processamento de dados de forma automática, com capacidade de processamento de dados em larga escala, lembrando que está atividade era realizada manualmente pelos pesquisadores, o que fazia com que o tratamento dos dados ocupasse uma grande parcela de tempo.

Além disso, o *workflow* se aproveita de uma base de dados pré-existente para produzir dados curados conforme o modelo apresentado e já utilizado pelo grupo de pesquisa. Além do workflow, desenvolvemos e testamos os módulos capazes de executar processamento paralelo. Também é importante ressaltar que houve ganhos de desempenho quando compara-se as soluções sequenciais e paralelas com diferentes cargas de dados.

A metodologia utilizada, que combina o uso *workflows* científicos com a computação paralela, é outra contribuição importante, pois mostra que esta metodologia é viável além de

demonstrar que existem ganhos de desempenho se comparados utilização de *workflows* usando somente métodos sequenciais.

5.3 Limitações do Estudo

Durante a fase de desenvolvimento dos *workflows* concretos, problemas técnicos impediram que os teste de desempenho fossem realizados na máquina paralela Coyote oferecida pelo PPGMMC (Programa de Pós-graduação em Modelagem Matemática e Computacional), houve falha dos discos e a máquina foi desabilitada durante longo período, oque atrasa as rodadas dos experimentos neste ambiente. Por isso, os testes foram conduzidos em uma máquina local com placa gráfica NVIDIA, com suporte a CUDA, o que a pesar de não representar a realidade de um ambiente distribuído com um servidor de grande porte, em nada impediu que a pesquisa continuasse e se desenvolvesse com êxito.

Uma limitação a ser mencionada é o fato de os *workflows* implementados utilizarem somente a regressão linear simples, que apesar de ser amplamente difundida no tratamento de dados meteorológicos, não é único método estatístico que deve fazer parte de uma solução para o problema apresentado. Ressalta-se que os mesmos *workflows* são capazes de aceitar novos métodos, bastando substituir ou acrescentar novos módulos de processamento estatístico.

Outro fato importante a ser citato é a utilização da computação paralela em somente um *workflow* da solução, naturalmente outras etapas desse experimento podem ser desenvolvidas, usando paralelismo, mas por conta do tempo de aprendizado das técnicas e a necessidade de pesquisa aprofundada sobre outros aspectos do problema, guiaram o desenvolvimento de uma parte específica do experimento contando com código paralelo em CUDA.

5.4 Trabalhos Futuros

Os primeiros resultados obtidos trabalho nesse não esgotam o tema, pelo contrário, através deles é possível listar algumas novas perspectivas para desdobramento desta pesquisa em trabalhos futuros, a saber:

Desenvolvimento de novas versões do *workflow*, com mais pontos de paralelismo, com por exemplo na comparação dos valores de mínimos e máximos definidos pelo pesquisador.

Implementação de outros modelos estatísticos mais sofisticados, como Ponderação, Regional (PR), Regressão Linear Múltipla, Regressão Linear/Ponderação Regional (RL/PR) que já estão em fase de estudos pelo grupo de pesquisa.

Criação de módulo de integração e exportação de dados, para o sistema *WebOntology* (BARBOSA; CRUZ, 2013), que é um sistema de gerenciamento semântico de dados meteorológicos apoiado por ontologias bem fundamentadas, desenvolvido pelo aluno de graduação Thiago Barbosa durante sua iniciação Científica, em parceria com o grupo de pesquisa na área de meteorologia da UFRRJ.

Expansão do *workflow* com o módulo exportação de dados no formato RDF, (COSTA; SILVA; CRUZ, 2013) desenvolvido pelo aluno de graduação Mário Junior Ribeiro da Costa durante sua iniciação Científica, em parceria com o grupo de pesquisa na área de meteorologia da UFRRJ.

REFERENCIAS

ALVES FILHO, A. P.; RIBEIRO, H. A percepção do caos urbano, as enchentes e as suas repercussões nas políticas públicas da Região Metropolitana de São Paulo. **Saúde e Sociedade**, v. 15, n. 3, p. 145–161, 2006.

ASVIJA B et al. Provisioning the MM5 meteorological model as Grid Scientific Workflow. 2010.

BARBOSA, T. M. DA S.; CRUZ, S. M. S. DA. **Uma Abordagem de Gerenciamento Semântico de Experimentos Meteorológicos em Pluviometria.pdf**, 2013. Disponível em: <https://docs.google.com/file/d/0ByMLHgMMHCODM2h0UFRPa0tiZEE/edit?usp=sharing>

BARBOSA, V. C. **An introduction to distributed algorithms [...] [...]**. Cambridge, Mass. [u.a.: MIT Press, 1996.

CHAKRABARTI, G. et al. CUDA: Compiling and optimizing for a GPU platform. **Procedia Computer Science**, v. 9, p. 1910–1919, jan. 2012.

COSTA, M. J. R. DA; SILVA, F. C. DA; CRUZ, S. M. S. **GERÊNCIA DE TRIPLAS RDF DE PROVENIÊNCIA EM EXPERIMENTOS CIENTÍFICOS EM LARGA ESCALA**. In: I REUNIÃO ANUAL DE INICIAÇÃO CIENTÍFICA DA UFRRJ - I RIAC,. 2013

CRUZ, S. M. S. DA. **UMA ESTRATÉGIA DE APOIO À GERÊNCIA DE DADOS DE PROVENIÊNCIA EM EXPERIMENTOS CIENTÍFICOS**. Rio de Janeiro - Brasil: Universidade Federal do Rio de Janeiro, 1 ago. 2011.

CUDA Toolkit Documentation. Disponível em: <http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#abstract>. Acesso em: 9 ago. 2013.

DA SILVA, V. P. et al. Análise da pluviometria e dias chuvosos na região Nordeste do Brasil. **R. Bras. Eng. Agríc. Ambiental**, v. 15, n. 2, p. 131–138, 2011.

DAVIDSON, S. B.; FREIRE, J. **Provenance and scientific workflows: challenges and opportunities**Proceedings of the 2008 ACM SIGMOD international conference on Management of data. **Anais**...2008Disponível em: <http://dl.acm.org/citation.cfm?id=1376772>. Acesso em: 15 jan. 2014

DE PAULA CORRÊA, M. A DIVULGAÇÃO DO ÍNDICE ULTRAVIOLETA COMO PREVENÇÃO AO EXCESSO DE EXPOSIÇÃO AO SOL: UMA CONTRIBUIÇÃO DA METEOROLOGIA PARA O DESENVOLVIMENTO DE POLÍTICAS PÚBLICAS PARA A SAÚDE NO PAÍS. 2005.

DEELMAN, E. et al. Workflows and e-Science: An overview of workflow system features and capabilities. **Future Generation Computer Systems**, v. 25, n. 5, p. 528–540, maio 2009.

DENNING, P. J.; TICHY, W. F. Highly Parallel Computation. 1990.

EAGLESON, P. S. The evolution of modern hydrology*(f ram watershed to continent in 30 years). 1993.

FENG, S.; HU, Q.; QIAN, W. Quality control of daily meteorological data in China, 1951–2000: a new dataset. **International Journal of Climatology**, v. 24, n. 7, p. 853–870, 15 jun. 2004.

FERRARI, G. T. **Imputação de dados pluviométricos e sua aplicação na modelagem de eventos extremos de seca agrícola**. Piracicaba: Universidade de São Paulo - Escola Superior de Agricultura "Luiz de Queiroz", 2011.

FILHO, G. R. L. et al. Assimilação, Controle de Qualidade e Análise de Dados de Meteorológicos Apoiados por Proveniência. p. 8, 2013.

FREIRE, J. et al. Provenance for computational tasks: A survey. **Computing in Science & Engineering**, v. 10, n. 3, p. 11–21, 2008.

HARRIS, M. **GPGPU.org**. Disponível em: <http://gpgpu.org/about>. Acesso em: 19 ago. 2013.

HEY, T.; TOLLE, K. **The fourth paradigm data-intensive scientific discovery**. Redmond, Wash.: Microsoft Research, 2009.

HONG, Z.; DA-FANG ZHANG; XIA-AN, B. Comparison and Analysis of GPGPU and Parallel Computing on Multi-Core CPU. **International Journal of Information and Education Technology**, 2012.

HORTA, F. et al. Prov-Vis: Large-Scale Scientific Data Visualization Using Provenance. 2013a.

HORTA, F. et al. **Provenance traces from Chiron parallel workflow engine**Proceedings of the Joint EDBT/ICDT 2013 Workshops. **Anais**...2013bDisponível em: <http://dl.acm.org/citation.cfm?id=2457379>. Acesso em: 16 dez. 2013

HULL, D. et al. Taverna: a tool for building and running workflows of services. **Nucleic Acids Research**, v. 34, n. Web Server, p. W729–W732, 1 jul. 2006.

INMET. Disponível em: <http://www.inmet.gov.br/portal/index.php?r=home/page&page=sm_previsao_tempo>. Acesso em: 24 ago. 2013a.

INMET. **NORMAIS CLIMATOLÓGICAS DO BRASIL, PERÍODO 1961-1990**, 2013b. Disponível em: <http://www.inmet.gov.br/webcdp/climatologia/normais/imagens/normais/textos/apresentacao.pdf>. Acesso em: 9 maio. 2014

JACOB, F. Modeling Parallel Programs for Heterogeneous Computing. 2010.

LASTOVETSKY, A. Adaptive parallel computing on heterogeneous networks with mpC. 2002.

LIMA, R. A. O.; SANTOS, R. P. DOS. **CONTROLE DE QUALIDADE DE DADOS METEOROLOGICOS**. [s.l: s.n.].

LIRA, M. A. T.; DA SILVA, E. M.; BRABO, J. M. Estimativa dos recursos eólicos no litoral cearense usando a teoria da regressão linear. **Revista Brasileira de Meteorologia**, v. 26, n. 3, p. 349–366, 2011.

LUDÄSCHER, B. et al. Scientific Workflow Management and the Kepler System. mar. 2005.

MAGINA, F. DE C. Aquisição Automática e Tratamento de Dados Meteorológicos Aplicáveis ao Projeto e Operação de Linhas Aéreas de Transmissão de Energia Elétrica. **Aquisição Automática e Tratamento de Dados Meteorológicos Aplicáveis ao Projeto e Operação de Linhas Aéreas de Transmissão de Energia Elétrica**, 2007.

MATTOSO, M. et al. **User-steering of HPC workflows: state-of-the-art and future directions**Proceedings of the 2nd ACM SIGMOD Workshop on Scalable Workflow Execution Engines and Technologies. **Anais**...2013Disponível em: <http://dl.acm.org/citation.cfm?id=2499900>. Acesso em: 16 dez. 2013

MELLO, M. P.; PETERNELLI, L. A. **Conhecendo o R Uma visão mais que Estatística**. [s.l.] Editora UFV, 2013.

myGrid. Disponível em: <http://www.mygrid.org.uk/>. Acesso em: 22 jun. 2014.

NICKOLLS, J. et al. Scalable parallel programming with CUDA. **Queue**, v. 6, n. 2, p. 40–53, 2008.

OKITSU, Y.; INO, F.; HAGIHARA, K. High-performance cone beam reconstruction using CUDA compatible GPUs. **Parallel Computing**, v. 36, n. 2-3, p. 129–141, fev. 2010.

OLIVEIRA, D. et al. Ontology-based Semi-automatic Workflow Composition. **Journal of Information and Data Management**, fev. 2012.

Open MPI: Open Source High Performance Computing. Disponível em: <http://www.open-mpi.org/>. Acesso em: 24 jun. 2014.

OpenMP.org. Disponível em: <http://openmp.org/wp/>. Acesso em: 24 jun. 2014.

PROV-DM: The PROV Data Model. Disponível em: <http://www.w3.org/TR/2013/REC-prov-dm-20130430/>. Acesso em: 26 maio. 2014.

SILVA, C. T. et al. **Using vistrails and provenance for teaching scientific visualization**Computer Graphics Forum. **Anais**...Wiley Online Library, 2011Disponível em: <http://onlinelibrary.wiley.com/doi/10.1111/j.1467-8659.2010.01830.x/full>. Acesso em: 22 jun. 2014

SIMERJ - Sistema de Meteorologia do Estado do Rio de Janeiro. Disponível em: <http://www.simerj.com/>. Acesso em: 21 jun. 2014.

TAN, W. et al. A comparison of using Taverna and BPEL in building scientific workflows: the case of caGrid. **Concurrency and Computation: Practice and Experience**, p. n/a–n/a, 2009.

TANENBAUM, A. S.; STEEN, M. V. **Distributed Systems Principles and Paradigms**. 2. ed. [s.l: s.n.].

TAURION, C. **Cloud Computing - Computação em Nuvem - Transformando o mundo da Tecnologia da Informação**. Disponível em: <http://books.google.com.br/books?id=mvir2X-A2mcC&printsec=frontcover&hl=pt-BR&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false>. Acesso em: 24 ago. 2013.

The Kepler Project — Kepler. Disponível em: <https://kepler-project.org/>. Acesso em: 29 jun. 2014.

TIWARI, A.; SEKHAR, A. K. T. Workflow based framework for life science informatics. **Computational Biology and Chemistry**, v. 31, n. 5-6, p. 305–319, out. 2007.

VAN DER AALST, W. M. . et al. **Workflows Patterns**, 2002.

VisTrails Documentation., 2013. Disponível em: <http://www.vistrails.org/usersguide/v2.0/html/VisTrails.pdf>. Acesso em: 16 ago. 2013