TIMESAT-Cluster: Uma Abordagem em Cluster para Analisar Séries-Temporais de Dados Produzidos por Sensores de Satélite

Clovis A. S. Silva[1], **Renato P. Ishii**[1], **Carlos R. Padovani**[2]

[1]Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
Caixa Postal 549 – 79.070-900 – Campo Grande – MS – Brasil

[2]Embrapa Pantanal –
Caixa Postal 109 – 79320-900 – Corumbá – MS – Brasil

renato@facom.ufms.br, ocrovis@gmail.com, carlos.padovani@embrapa.br

Abstract. *Currently exists a large amount of information available, coming from different fonts, resulting in the onset of some problems mainly related to the processing and storage of this growing volume of information. One of these data sources has aroused interest in scientific community who are the satellite sensors. These sensors produce images at different levels (or layers) and allows analysis in several large areas of knowledge. In this scenario arises some questions: how to perform the data processing in an environment where infrastructure isn't a limiting? How to store this information? Many times the user doesn't have available all the resources, such as computers robust and space for storage. This article describes in details the TIMESAT-Cluster, a platform that extends the features of the original TIMESAT , allowing their execution in a high performance environment, or in a cluster of computers. This platform has a Web interface that provides access to the main functions of TIMESAT to analyze time series originating from satellite images. Experimental results demonstrate the reduction in run-time six times when compared to the original TIMESAT.*

Resumo. *Existe atualmente uma grande quantidade de informação disponível, oriundas de fontes diversas, ocasionando o surgimento de alguns problemas relacionados, principalmente, ao processamento e ao armazenamento deste volume crescente de informação. Uma dessas fontes de dados tem despertado interesse na comunidade científica que são os sensores de satélite. Tais sensores produzem imagens em diferentes níveis (ou camadas) e permitem análises nas mais diversas áreas do conhecimento. A partir deste cenário surgem algumas questões: como realizar o processamento desses dados em um ambiente onde a infraestrutura não é um limitante? Como armazenar essa informação? Muitas vezes o usuário não tem disponível todos os recursos, tais como computadores robustos e espaço para armazenamento. Este artigo descreve em detalhes o TIMESAT-Cluster, uma plataforma que estende as funcionalidades originais do TIMESAT, permitindo sua execução em uma ambiente de alto desempenho, ou seja, em um cluster de computadores. Além disso, esta plataforma fornece uma interface Web, que provê acesso às principais funções do TIMESAT a fim de analisar séries temporais originárias de imagens de satélites. Resultados experimentais demonstram a redução do tempo de execução em seis vezes,*

quando comparado ao TIMESAT original.

1. Introdução

Imagens produzidas por satélites são importantes fontes de dados para análises de informações sobre determinadas regiões, que podem ser extraídas e identificadas a partir de estudos e análises dessas imagens. Inundações, alagamentos, processos de desertificação, queimadas e desmatamentos são algumas das características que podem ser extraídas das áreas investigadas.

Evidentemente que as análises das imagens, oriundas de áreas específicas, não podem ser realizadas apenas de forma empírica. Portanto, é comum nesta área de pesquisa, utilizar ferramentas automatizadas que processam e fornecem informações mais detalhadas sobre a região em estudo.

A ferramenta utilizada para processar as imagens de satélite neste trabalho é o *TIMESAT* [Eklundh and Jönsson 2015], um software cujas principais funções são de suavizar e extrair informações de séries temporais a partir de uma sequência de imagens durante um período determinado de tempo.

As séries temporais são "coleções de dados realizados em sequência durante um período de tempo" [Ehlers 2007]. Elas são geradas pelo *TIMESAT* a partir das sequências de imagens por satélite, e então são filtradas. Este processo realizado pelo *TIMESAT* constitui-se da eliminação de ruído e da suavização das séries temporais utilizando alguns dos seguintes métodos de filtragem: Savitzky-Golay, Gaussiano Assimétrico e Logístico duplo [Eklundh and Jönsson 2015].

O uso do *TIMESAT* é positivo para as análises de imagens por satélite, mas na abordagem tradicional existem alguns problemas, tais como alto custo de processamento para dados massivos, além da dificuldade de armazenamento para grandes quantidades de dados. A partir dessa limitação, foi desenvolvida uma ferramenta para auxiliar pesquisadores a utilizar o *TIMESAT* de forma simplificada e reduzir o seu tempo de execução. Esta ferramenta, denominada *TIMESAT-Cluster*, utiliza *clusters*[1] para o processamento e armazenamento de dados fornecidos. Além de diminuir o tempo de processamento, ela possui características tais como boa usabilidade e fácil acesso por meio da *Web* em diferentes dispositivos.

A fim de validar o *TIMESAT-Cluster*, foram realizados experimentos com imagens da região do Pantanal, fornecidas pela Embrapa-Patanal[2], para analisar alagamentos e outras características dessa região ao longo de um período de tempo. Observando os resultados obtidos a partir desses experimentos, verifica-se que o tempo de execução do *TIMESAT* diminuiu de aproximadamente alguns dias para apenas algumas horas.

Este trabalho está organizado como segue: a seção 2 descreve alguns trabalhos e produtos relacionados ao *TIMESAT-Cluster*. Na seção 3 são descritas todas as tecnologias empregadas e detalha a forma como foi desenvolvida a interface Web e a abordagem em cluster utilizada neste trabalho. A discussão sobre os resultados é apresentada na seção 4 e por fim a conclusão está apresentada na seção 5, bem como propostas de trabalhos futuros são discutidas.

[1]Um cluster consiste de um aglomerado de computadores interligados por meio de uma rede de comunicação, que são vistos de forma transparente, ou seja, visto como se fossem apenas um único computador.

[2]Site institucional da Embrapa Pantanal `https://www.embrapa.br/pantanal`

2. Trabalhos relacionados

O *TIMESAT* é uma ferramenta muito utilizada em trabalhos nas áreas de geoprocessamento e de sensoriamento remoto com a finalidade de melhorar as análises de regiões pesquisadas. O estudo [Borges 2014] por exemplo, desenvolvido para mapear a cobertura vegetal do Oeste da Bahia, utiliza os métodos de suavização disponíveis no *TIMESAT*, a fim de encontrar nele qual desses métodos se adapta melhor as séries temporais analisadas.

Outros trabalhos descrevem um modelo baseado em *cluster* ou nuvem para resolver problemas computacionais, como o [Yang et al. 2012], que fundamenta de forma detalhada como utilizar serviços em nuvem para processar dados massivos. No trabalho proposto, os pesquisadores mostram quais os padrões utilizados para integrar e fazer a comunicação entre os diferentes serviços utilizados. Além disso, mostram todo o processo de implementação e execução do ambiente. Uma das abordagens importantes do trabalho está relacionado ao armazenamento de dados, que foi solucionado usando o serviço da *Amazon* denominado *Amazon S3*. Outro trabalho que pode ser citado é o *Google Cloud Vision API* [GoogleCloudVisionAPI], um serviço que devolve informações e características sobre o conteúdo de uma imagem.

É evidente que existem diferenças nas abordagens e métodos dos trabalhos citados a este apresentado. O *Google Cloud Vision* por exemplo, funciona como um serviço em nuvem sob demanda que possui características fundamentais como a escalabilidade, transparência e multiplos clientes. Por outro lado, a abordagem proposta neste trabalho integra *clusters* específicos com a finalidade de executar o processamento do *TIMESAT*, sem a necessidade de fornecer serviços a múltiplos clientes de forma transparente, mas sim a clientes específicos.

3. Abordagem proposta

O desenvolvimento do *TIMESAT-Cluster* consistiu-se da união de três tecnologias computacionais principais: uma aplicação *Web*, o *TIMESAT* e um ambiente de cluster. A Figura 1 apresenta, de maneira simplificada, um esquema que representa a integração das três tecnologias utilizadas neste trabalho. O *TIMESAT-Cluster* é um sistema Web que fornece a interface para o gerenciamento de dados no *cluster*, onde está instalado o *Rocks Cluster*. O *TIMESAT-Cluster* também é responsável pela execução do *TIMESAT* em algum *host* do *cluster*. Os detalhes desse processo de execução e o relacionamento entre as tecnologias é discutido na seção 3.4.

3.1. *TIMESAT – A Software Package to Analyse Time-Series of Satellite Sensor Data*

O *TIMESAT*, como apresentado previamente, é um software cuja principal função é a de suavizar séries temporais. Ele possui três métodos de filtragem: Savitzky-Golay, Gaussiano Assimétrico e Logístico duplo [Eklundh and Jönsson 2015]. Além disso, é importante destacar que após o processamento, o *TIMESAT* fornece como resultado alguns parâmetros de sazonalidade, tais como início de um ciclo na série temporal, fim de um ciclo, duração de um ciclo, amplitude, dentre outros. Portanto, o software fornece um conjunto de informações fundamentais para analisar com maior detalhamento informações dessas séries. As funcionalidades do *TIMESAT* não se limitam apenas a isso, mas neste trabalho são abordados apenas suas características gerais de execução. Para mais detalhes sobre o *TIMESAT*, consulte o seu manual [Eklundh and Jönsson 2015].

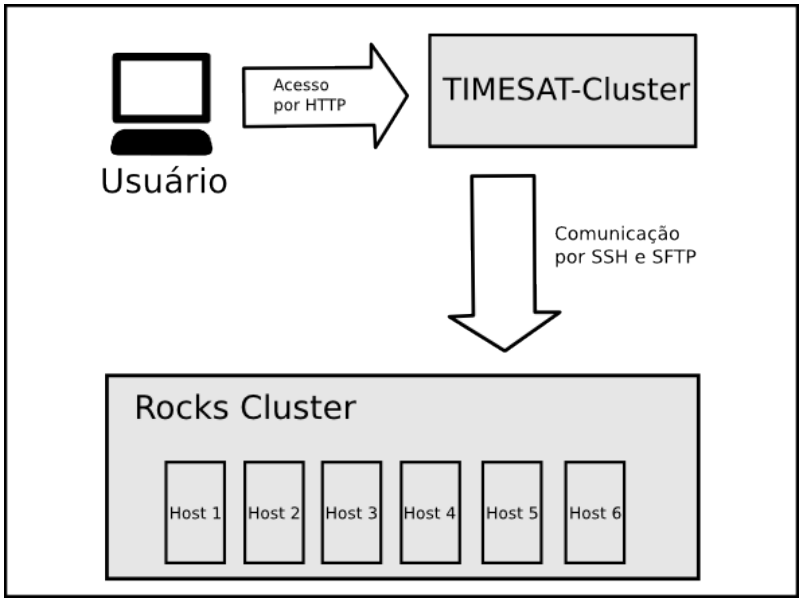

Figura 1. Esquema simplificado para ilustrar de que forma há a comunicação entre as principais tecnologias utilizadas no *TIMESAT-Cluster*.

3.2. Infraestrutura em *Cluster*

Clusters da maneira como foi abordado neste trabalho são aglomerados de computadores interligados por uma rede de comunicação, que são vistos como um único computador de forma transparente. A infraestrutura utilizada nesta abordagem depende de um *cluster* para integrar-se a outras tecnologias. Utilizou-se então sete computadores interligados: um representando o *frontend* e os outros seis como *hosts*, gerenciados pelo sistema operacional *Rocks Cluster* [3]. O *Rocks Cluster* é uma distribuição *Linux* voltada para computação de alto desempenho em *clusters*. Além de ser simples de instalá-la e utilizá-la, possui recursos que facilitam muito o seu gerenciamento, simplificando assim o desenvolvimento de ferramentas que utilizam o *cluster* para processar algum tipo de informação. Um software em específico destaca-se nessa distribuição, o *Ganglia* [4], responsável pelo monitoramento do *cluster* que provê uma quantidade extensa de informações tais como utilização de discos rígidos e consumo de memória *RAM* e *CPUs*, além de métricas a respeito de todos os *hosts* do *cluster*. Durante o processo de execução do *TIMESAT* no modelo proposto neste trabalho, o monitoramento da execução é realizado por meio do *Ganglia*, sendo possível assim identificar quaisquer tipos de anomalia no processo de análise das séries temporais. Para o gerenciamento das funções do *cluster* destacam-se alguns comandos descritos a seguir:

rocks run host Executar um comando em um *host* específico.
rocks sync config Reconfigurar cada um dos arquivos de configurações e reiniciar os serviços relevantes.
rocks sync host firewall Reconfigurar e reiniciar o *firewall* em um determinado *host*.

[3]O projeto Rocks Cluster está acessível através do endereço eletrônico - `http://www.rocksclusters.org/`

[4]O projeto está acessível através do endereço eletrônico `ganglia.sourceforge.net`

rocks list host Listar os membros, número de *CPUs* e posição física dentro de uma lista de *hosts*.

rocks add host Adicionar um novo *host* para o *cluster*.

A lista completa de funcionalidades está disponível em [5]

3.3. A interface *Web - TIMESAT-Cluster*

Nesta etapa foi desenvolvida uma interface que facilitasse o uso e execução do *TIMESAT*. Optou-se então por utilizar a *Web* como ambiente de desenvolvimento, devido a praticidade e a simplicidade na criação de aplicativos. Foi utilizado também o popular *Framework Ruby on Rails* [6] que destaca-se pela sua facilidade de codificação e de construção de aplicações *Web* flexíveis.

Algumas características mais peculiares do software devem ser destacadas. O processo de *upload* de imagens é realizado utilizando o protocolo *SFTP* [7], com as gems *Paperclip* [8] e *Net-SFTP* [9]. Esse modelo foi escolhido para ser utilizado pois os arquivos precisavam ser armazenados com segurança no ambiente de *clusters*. A autenticação e a segurança foram implementadas com a gem *Devise*[10], economizando assim um grande tempo de desenvolvimento, uma vez que seus recursos e funcionalidades são facilmente integráveis ao sistema desenvolvido.

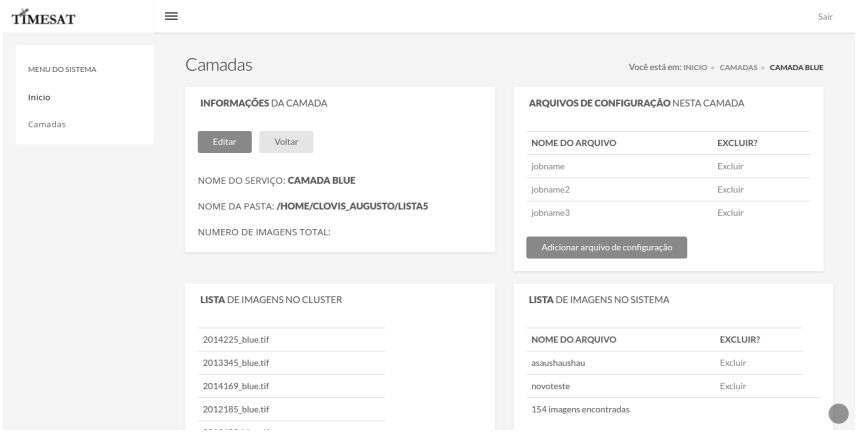

Figura 2. Tela da interface Web do *TIMESAT-Cluster* com informações de configurações gerais e lista de imagens enviadas ao sistema.

Na figura 2, na página inicial de configuração do sistema, destacam-se as camadas oriundas dos diferentes sensores de um satélite. Cada camada criada no sistema contempla informações de configuração que devem ser adicionadas antes do processamento. Uma camada deve possuir um nome, uma lista de imagens e um ou mais arquivos de configuração. A lista de imagens é inserida no sistema por meio do processo de *upload*. A

[5]A lista completa de funcionalidades está disponivel através do endereço - `http://central6.rocksclusters.org/roll-documentation/base/6.1.1/c2310.html`

[6]Framework Ruby On Rails disponível através do endereço `http://rubyonrails.org/`

[7]SFTP - SSH file transfer protocol`http://tools.ietf.org/html/draft-moonesamy-secsh-filexfer-00#page-4`

[8]Paperclip - gem para Ruby On Rails - `https://github.com/thoughtbot/paperclip`

[9]Net-SFTP - gem para Ruby On Rails - `https://github.com/net-ssh/net-sftp`

[10]Devise - gem para Ruby On Rails - `https://github.com/plataformatec/devise`

grande diferença dos métodos tradicionais é que no *TIMESAT-Cluster* ao adicionar uma das imagens da camada criada, elas são inseridas diretamente no *cluster* através do protocolo *SFTP*, como visto na figura 3. Com essa abordagem, soluciona-se o problema de falta de espaço para armazenamento, uma vez que a infraestrutura em *cluster* disponibilizada possui muito mais capacidade para armazenar arquivos que a versão original do *TIMESAT*.

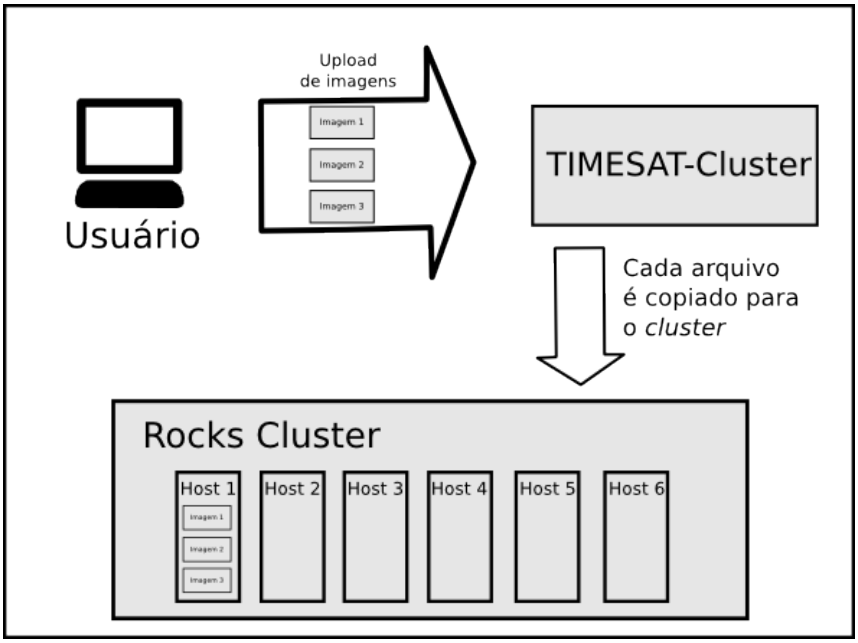

Figura 3. Esquema do processo de *upload* utilizando o *cluster* na abordagem proposta.

3.4. Uso do *TIMESAT-Cluster*

A necessidade de se construir uma solução simples, fez com que todas as funcionalidades do software proposto fossem criadas sempre partindo desse princípio. Logo, para executar o *TIMESAT* através do *TIMESAT-Cluster* basta seguir alguns poucos passos básicos:

1. Autenticar-se no sistema com email e senha;
2. Criar uma nova camada;
    - Enviar ao sistema todas as imagens necessárias para a camada;
    - Criar no mínimo um arquivo de configuração para a camada;
        - Por fim, executar um novo processo a partir do arquivo de configuração criado;
3. Realizar o "download"dos arquivos gerados pelo processamento.

Dos passos citados acima, apenas os de execução e de *download* ainda não foram mencionados neste trabalho. O passo de execução é o ponto crucial do *TIMESAT-Cluster*. Nele são utilizadas de forma transparente e integrada todas as tecnologias até aqui citadas: *cluster*, *TIMESAT* e o próprio *TIMESAT-Cluster*.
Ao selecionar um arquivo de configuração para uma determinada camada e executá-la, uma nova instância do *TIMESAT* é iniciada em algum *host* do *cluster*. Esse modelo adotado, permite com que o *TIMESAT* seja executado de forma simultânea ou paralela nos n

hosts do *cluster*.
No ambiente de *cluster*, ao tentar iniciar um novo processamento a partir do *TIMESAT-Cluster*, o sistema seleciona qual dos *hosts* não está executando nenhuma instância do *TIMESAT*. Ao identificar e selecionar o *host* disponível, o sistema inicia nele uma nova instância do *TIMESAT* com as configurações selecionadas. Caso nenhum dos hosts esteja disponível, é exibida uma mensagem alertando o usuário e nenhuma ação é tomada, sendo necessário esperar o término de todos os processos em execução ou cancelar algum deles. Como a execução não é instantânea, o processamento solicitado é executado sem bloquear o *TIMESAT-Cluster*. Ao finalizar o processamento, é possível realizar o *download* de todos os arquivos gerados, tanto dos arquivos de configurações quanto as imagens processadas.

4. Resultados e discussão

O tempo de execução do *TIMESAT* no modelo tradicional, utilizando os dados fornecidos pelos pesquisadores da Embrapa-Pantanal[11], tinha um custo de alguns dias, pois cada sequência de imagens que representa uma camada era processada sequencial e individualmente. Com o novo modelo proposto, conseguiu-se executar várias instâncias do *TIMESAT* e com isso diminuir muito a complexidade e o tempo de execução.
Ao utilizar nos experimentos, um modelo de *cluster* com seis *hosts*, gerenciados pelo *Rocks Cluster* e pelo *TIMESAT-Cluster*, obteve-se bons resultados ao executar todas as seis camadas de forma paralela. Em uma abordagem teórica teríamos um *speedup* ideal de seis vezes em relação ao modelo tradicional. É evidente que o ganho ocorre por ter-se solucionado o principal problema: a falta de infraestrutura. Além disso, ainda foi criado um modelo que facilita a utlização e execução do *TIMESAT* para uma quantidade massiva de dados, o que mostra o quanto a execução de softwares em *clusters* ou em *nuvem* podem ajudar determinados setores da sociedade.

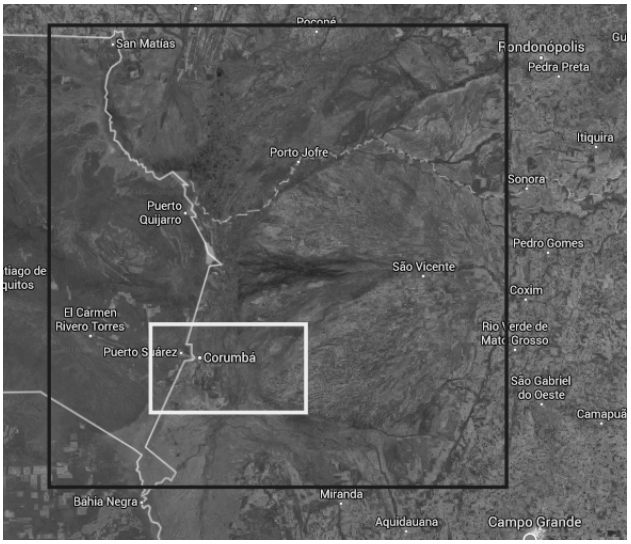

Figura 4. Imagem extraída do Google Maps para ilustrar as regiões processadas nos experimentos realizados.

[11] `https://www.embrapa.br/pantanal`

Os arquivos utilizados para os experimentos são divididos em seis diferentes camadas: Blue, NDVI, Red, EVI, MIR e NIR. Cada uma dessas camadas possui exatamente 138 arquivos de imagens produzidas por satélites, da ampla região do pantanal, onde cada camada possui 5.2GB e totalizando assim 31.2GB de dados para serem processados.
A fim de de conseguir comparar diferentes tempos de execução, foram utilizadas duas configurações para processar sub-áreas das imagens fornecidas. A primeira em uma região mais próxima a cidade de Corumbá, que pode ser vista na figura 4, com um contorno mais claro, e a segunda configuração em uma área mais ampla que abrange toda a região Pantaneira e pode ser visualizada com um contorno mais escuro na figura 4.
Na figura 5, pode-se analisar os tempos e observar que duas camadas "EVI"e "MIR"são as que mais demandam tempo para serem processadas. Essa diferença já não ocorre no gráfico da figura 6, em razão da área processada ser muito maior, nivelando dessa forma os tempos.

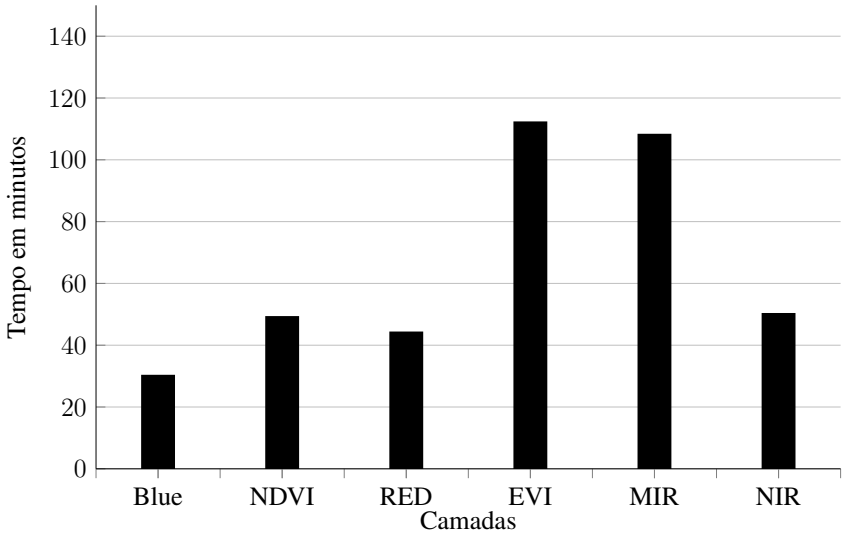

Figura 5. Tempo de execução de uma pequena área da imagem para cada camada no *cluster* - Região próximo a cidade de Corumbá

Em termos práticos, para experimentos executados pode-se realizar cálculos para *speedup*. No primeiro gráfico apresentado na figura 5 em abordagem tradicional o tempo de execução $t1$ de todos os processos em todas as camadas foi de 393 minutos, já na abordagem em *cluster* o tempo total $t2$ foi de 112 minutos.

$$speedup = \frac{t1}{t2} = \frac{393}{112} = 3.50$$

A melhoria no tempo de execução na primeira sub-área foi de 3.50 vezes em relação a abordagem tradicional. No segundo gráfico da figura 6, o cálculo de speedup é dado por:

$$speedup = \frac{4522}{847} = 5.34$$

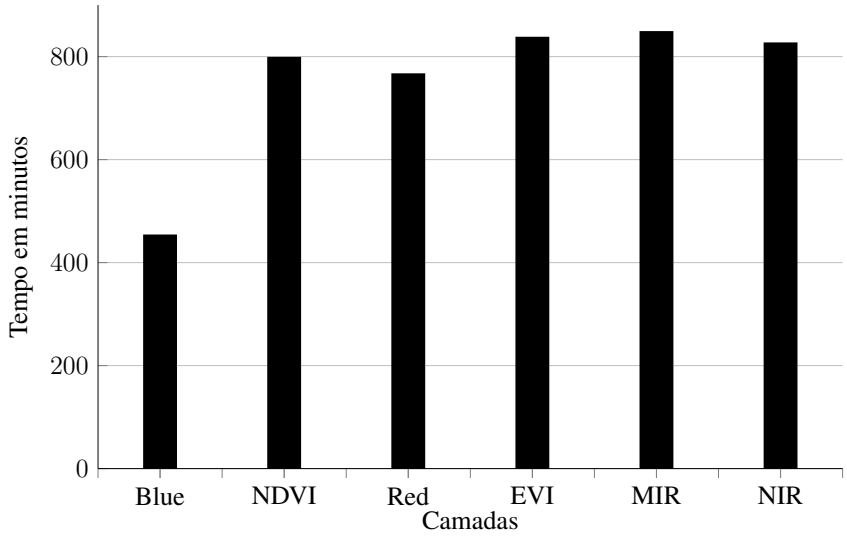

Figura 6. Tempo de execução de uma grande área da imagem para cada camada no *cluster* - Região que abrange todo o Pantanal

Nesse caso a melhoria foi de 5.34 vezes, se aproximando do *speedup* ideal de seis vezes.

5. Conclusão

Este trabalho demonstrou a utilidade de *clusters* para execução de softwares onde existe uma grande demanda por espaço e por processamento. Além claro, de mostrar a possibilidade de criar aplicações que utilize de forma eficiente esse tipo de abordagem. Demonstra-se também a flexibilidade ao se utilizar a *Web* como meio de transmissão de dados e como ambiente de desenvolvimento.

Algumas dificuldades foram encontradas durante o projeto. A própria limitação da *Web* é um dos fatores que dificultam a criação de um aplicativo para processar uma grande quantidade de dados devido a fatores como a transferência de dados. Imagine o gargalo que é, por exemplo, fazer "upload"de 30GB de dados.

O fato do *TIMESAT* não ter o código aberto e disponível para modificação e melhorias também limitou o desenvolvimento da aplicação e dificultou a integração com a interface Web, além de tornar impossível qualquer tipo de alteração em suas funcionalidades.

Para o futuro, alguns pontos serão desenvolvidos:

- Melhorias ao gerar os arquivos de configurações para cada camada;
- Caso o código do *TIMESAT* seja liberado para mudanças em breve, o processo de integração com o aplicativo desenvolvido será muito mais profundo, já que será possível ter o acesso a algumas operações que o software executa;
- Melhorias no escalonamento dos processos no cluster, ou seja, executar uma instância do *TIMESAT* apenas quando o nó do cluster não estiver sendo usado por nenhum outro processo que exija muito processamento;
- Visualização do dados gerados no *TIMESAT* pelo próprio aplicativo *Web*. Esta última, seria uma grande melhoria, já que o usuário não necessitaria fazer "download"dos arquivos processados para a sua máquina local e então vizualizá-los por um software específico. Esse processo de visualização ocorreria no próprio aplicativo *Web*.

Referências

Borges, Elane Fiúza e Sano, E. E. (2014). Séries temporais de evi do modis para o mapeamento de uso e cobertura vegetal do oeste da bahia/temporal series of evi from modis sensor for land use and land cover mapping of western bahia. *Boletim de Ciências Geodésicas*, 20(3):526.

Ehlers, R. S. (2007). Análise de séries temporais. *Laboratório de Estatística e Geoinformação. Universidade Federal do Paraná*.

Eklundh, L. and Jönsson, P. (2015). *Remote Sensing Time Series: Revealing Land Surface Dynamics*, chapter TIMESAT: A Software Package for Time-Series Processing and Assessment of Vegetation Dynamics, pages 141–158. Springer International Publishing, Cham.

Eklundh, L. and Jönsson, P. (2015). Timesat 3.2 software manual, lund and malmö university, sweden.

GoogleCloudVisionAPI. Google Cloud Vision API. `https://cloud.google.com/vision/`. [Accessado em 12 de Março de 2016].

Yang, C., Shao, Y., Chen, N., and Di, L. (2012). The cloud computing for a dynamic agro-geoinformation processing. In *Agro-Geoinformatics (Agro-Geoinformatics), 2012 First International Conference on*, pages 1–4. IEEE.