View Reviews

Paper ID	435
Paper Title	Massively-Parallel Change Detection for Satellite Time Series Data with Missing Values
Track Name	Research Paper Second-Round

Reviewer #1

Questions

1. Overall Evaluation
Accept

2. Confidence
Knowledgeable

3. Originality
Medium

4. Importance
High

5. Summary of the contribution (in a few sentences)
The idea of the paper is to parallelize the change detection in a time series data so that it is scalable and can also be used on parallel CPUs of a commodity machine. The paper parallelizes an existing unsupervised learning algorithm called BFAST-Monitor (break detection for additive season and trend) to achieve this and also applies it on a deforestation detection use case over the entire continent of Africa to demonstrate the scalability of this massive parallelization technique. This approach shows up to 3 orders of magnitude performance improvement over several baselines.

6. List three or more strong points, labelled S1, S2, S3, etc.
S1. The paper shows the importance of parallelizing unsupervised learning techniques such as regression based BFAST over supervised deep learning techniques that can potentially take a long time to train. In this context, it explains how the regression based technique works by fitting the time series to a stable time and then focusing on the possible changes that could occur during a monitoring time. The combination of these two steps allows the linear regression technique to detect anomalies in the time series in an unsupervised manner. This is clearly explained in the paper and demonstrates the need to resort to simpler learning models in an era where deep learning is used in almost every application.

S2. The paper also explains the non-triviality of parallelizing in cases of missing data values (such as deforestation readings missing because of clouds obstructing the satellite's readings).

S3. The paper contrasts its batch-wised parallelization techniques for matrix multiplication and inversion against two extreme parallelism techniques that are naïve implementations that parallelize the per-pixel computations. Such naïve implementations lose the benefit of cache temporality and lead to under utilization of hardware when the batch is not large enough. Thus, the paper presents its batched parallelization technique thus retaining the batches that are accessed by all the threads in the shared memory.

S4. The paper conducts its experiments against several baselines – inner parallelism always sequentialized with tiled matrix multiplication computations (RgTI-EfSeq), only outer parallelism exploitation for matrix multiplication (Full-EfSeq) and a hand-written parallel C implementation in OpenMP (C). In fact the GPU accelerated parallelized BFAST algorithm shows up to 24 x and 48 x improvement over the OpenMP C implementation two artificial datasets (D5 and D6) used in the paper. On other baselines, the proposed approach demonstrates up to 5000x improvement over the traditional R implementation on the Peru (large) dataset which takes over 25 hours to process by the baseline R implementation. Similar impressive results are also shown on the real-world Peru(small) and Africa datasets.

7. List three or more weak points, labelled W1, W2, W3, etc.

W1. The paper does not inform the reader enough about the semantic significance of the matrix operations such as matrix multiplication and inversion for the time change detection algorithm.

W2. On similar lines as W1, while explaining the algorithm 1 in section 3a, the paper needs to elaborate upon how the matrix operations are the only bottleneck for the computation of moving sum, because the remaining paper is dedicated to parallelizing the matrix operations. May be a couple of real world examples here demonstrating how the matrices such as M inverse are computed in practice and accompanying them with statistics of processing times may help the reader prepare for the proposed solution in section 3b.

8. Detailed evaluation, labelled D1, D2, D3 etc.

Shared memory based parallelism using batch-wise partitioning of data is not a radically new concept. It is interesting that such a simple and well known idea has brought impressive results. The paper may want to mention if no such ideas were presented by earlier works and may add a discussion section contemplating its design choices and contrasting them with others to justify why their solution works towards the end of the experimental section.

Overall this paper not only solves an important and a hard problem and its evaluation supports its claim with some impressive results that are reported. The paper lacks a bit of a connection in writing to prepare the reader before section 3b. This can be fixed based on the suggestions in W1b.

10. Candidate for a Revision? (Please note that the authors have only one month to revise)

No

15. Recommended decision for the revised paper (fill in only when reviewing revised papers)

Not Applicable

Reviewer #2

Questions

1. Overall Evaluation

Weak Accept

2. Confidence

Some Familiarity

3. Originality

Medium

4. Importance

Medium

5. Summary of the contribution (in a few sentences)

This paper implements BFAST-Monitor, a change detection method for satellite data in a parallel fashion. This implementation is tolerant to missing values within time series. It includes several performance optimisations that result in significant speed-ups at both individual kernel and application levels, with a faster GPU implementation. This paper finally provides analysis of the transformations on the kernel level and the complication strategy using artificial datasets, and illustrates the benefits of the GPU implementation using real-world data, varying some of the algorithm parameters.

6. List three or more strong points, labelled S1, S2, S3, etc.

S1: The proposed work presents an efficient highly parallel implementation of BFAST-Monitor whose efficiency looks promising.
S2. The method and the optimisations are well evaluated on synthetic and real-world datasets
S3: The paper is well structured and the approach is well explained

7. List three or more weak points, labelled W1, W2, W3, etc.

W1: The related work does not cover the state-of-the-art methods
W2: The length of the time series in the experiments is relatively short and this choice is not justified
W3: No real world dataset was used in the performance analysis

8. Detailed evaluation, labelled D1, D2, D3 etc.

D1: BFAST-Monitor is one of many unsupervised change detection methods that are not cited. It would be interesting to see how its performance compares with the other change detection methods. The authors refer to Zhu et al. [3] for comparison of different change detection methods but do not compare their new parallel implementation with other change detection methods.
D2: This paper analyses the performance of the solution on artificial datasets. While it is reasonable that the size and the characteristics need to be controlled, it is, however, difficult to verify the performance of the data in real-world applications.
D3: The length of the time series in the performance analysis is quite short (maximally a length of 1024 points) compared to existing time series datasets, this choice is not justified.

10. Candidate for a Revision? (Please note that the authors have only one month to revise)

Yes

11. Required changes for a revision, if applicable. Labelled R1, R2, R3, etc

R1: Please cite the state-of-the-art change detection methods and either include them in the experimental comparison or justify why they are not relevant to the present work.

Reviewer #3

Questions

1. Overall Evaluation

Weak Reject

2. Confidence

Some Familiarity

3. Originality

Medium

4. Importance

Medium

5. Summary of the contribution (in a few sentences)

The paper focuses on solving the problem of change detection in satellite images over a period of time in a scalable and efficient way. It particularly focuses on the case of change detection in vegetation/tree cover. It proposes a parallel version of BFAST algorithm (break detection for additive season and trend), which can be run on GPUs and is much faster than the current naive single machine implementation. The algorithm is parallelized by dividing the per pixel time series into batches and then running the operations on these batches individually in parallel. It is further optimized by reducing the number of disk accesses due to the choice of batch size.

6. List three or more strong points, labelled S1, S2, S3, etc.

S1: The experiments show significant speedup over baselines and other approaches of optimizations.

S2: The paper explains in a good detail the proposed optimizations.

S3: The paper provides a GPU implementation.

7. List three or more weak points, labelled W1, W2, W3, etc.

W1. Not clear why a pixel-level parallelization is not enough for the problem.

W2. Some examples can help in understandng the proposed method.

W3. Figures 9 and 11 seems to be irrelevant.

W4. The datasets used in the experiments are very small.

8. Detailed evaluation, labelled D1, D2, D3 etc.

D1. The paper uses BFast as a baseline for comparison and motivation. The paper claims that BFast cannot scale well as it cannot be efficiently parallelized. However, according to Figure 4, BFast can be parallelized at the pixel level, i.e., pixels can be processed in parallel. It is not clear why this is not enough given that the satellite data has at least hundreds of millions of pixels.

D2. It could help to add an example that explains how the BFast algorithm is limited and describes the key idea of the proposed algorithm.

D3. Since the paper focuses mainly on parallelizing the BFast algorithm without really affecting its accuracy, Figures 9 and 11 seems to be irrelevant to the proposed work. In other words, the baseline would produce the same exact maps as the ones in those two figures.

D4. The introduction motivates the problem when the input has hundreds of billions of pixels. However, the experiment section runs only on a dataset of around 100,000 pixels which is too small. The paper uses the vegetation data as a case study. The publicly available MODIS vegetation data at 250-meter resolution contains around seven billion pixels per day. How does the proposed method scale to that data?

10. Candidate for a Revision? (Please note that the authors have only one month to revise)

Yes

11. Required changes for a revision, if applicable. Labelled R1, R2, R3, etc

R1. Explain why pixel-level parallelization is not applicable. The original BFAST algorithm applies on a time series at the pixel level and hence can be massively parallelized. Notice that the complex steps that the paper optimizes (the matrix multiplication and inverse) are applied on the static X matrix which is not related to the input time series values. So, these two steps can be computed once before even the input pixels are read. With missing values, there are up-to n variations of the X matrix which are all static can be processed once in a one-time preprocessing phase.

R2. Explain the colon notation which is used heavily with matrices and vectors.