Information Sciences xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Towards building a data-intensive index for big data computing – A case study of Remote Sensing data processing

Yan Ma [a], Lizhe Wang [a,*], Peng Liu [a], Rajiv Ranjan [b]

[a] *Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, PR China*
[b] *Computational Informatics, CSIRO, Australia*

ARTICLE INFO

Article history:
Received 20 June 2014
Received in revised form 2 October 2014
Accepted 5 October 2014
Available online xxxx

Keywords:
Big data
Parallel computing
Data-intensive computing
Remote Sensing data processing

ABSTRACT

With the recent advances in Remote Sensing (RS) techniques, continuous Earth Observation is generating tremendous volume of RS data. The proliferation of RS data is revolutionizing the way in which RS data are processed and understood. Data with higher dimensionality, as well as the increasing requirement for real-time processing capabilities, have also given rise to the challenging issue of “Data-Intensive (DI) Computing”. However, how to properly identify and qualify the DI issue remains a significant problem that is worth exploring. DI computing is a complex issue. While the huge data volume may be one of the reasons for this, some other factors could also be important. In this paper, we propose an empirical model (DI_{RS}) of DI index to estimate RS applications. DI_{RS} here is a novel empirical model (DI_{RS}) that could quantify the DI issues in RS data processing with a normalized DI index. Through experimental analysis of the typical algorithms across the whole RS data processing flow, we identify the key factors that affect the DI issues mostly. Finally, combined with the empirical knowledge of domain experts, we formulate DI_{RS} model to describe the correlations between the key factors and DI index. By virtue of experimental validation on more selected RS applications, we have found that DI_{RS} model is an easy but promising approach.

1. Introduction

The advent of the high-resolution global Earth Observation era is revolutionizing the way in which Remote Sensing (RS) data are generated, processed, and analyzed. The latest generation of space-borne sensors are generating nearly continuous streams of massive RS imageries. These data streams are transmitted through downlink channels at a rate of several gigabits per second. The amount of RS data acquired by a single satellite data center is dramatically increasing by several TB per day [25,12]. The advances in the spatial, temporal, and spectral resolution of sensors also lead to the high dimensionality of RS imagery pixels. Meanwhile, large-scale RS applications [10,31,27,9,19,30] are exploiting multi-temporal RS data with regional to global coverage as input for processing. Obviously, Remote Sensing applications are overwhelmed with the amount of high-dimensional RS data.

With the proliferation of data, Remote Sensing data processing turns out to be extremely challenging. Issues to be considered include efficient management and rapid processing of massive RS data, pixels of high dimensionality, and especially various multi-staged algorithms of RS applications with higher complexity and intensive irregular data access patterns [21].

* Corresponding author.
E-mail address: Lizhe.Wang@gmail.com (L. Wang).

http://dx.doi.org/10.1016/j.ins.2014.10.006
0020-0255/© 2014 Elsevier Inc. All rights reserved.

The situation has become even worse because of the increasing real-time or near-real-time processing requirements of many time-critical applications like hazard monitoring and tracking [11,24]. Generally, Remote Sensing data processing, especially for large-scale environmental monitoring and research, is regarded as typical Data-Intensive (DI) problems [23].

DI issues occur when the volumes and rates of data emerge as the rate-limiting factor of processing, and promise a revolutionary change in the way we seek solutions [18,14]. However, most of the existing DI issues are qualitatively defined and tend to focus on problems related to the huge amount of data [4]. Relying on these obscure and qualitative definitions to determine whether a problem is data-intensive is rather difficult. For a deeper insight into DI problems, it is critical important to quantify the meaning of data intensiveness beyond a qualitative framework. The requirements and challenges for DI problems are totally different from those related to traditional computing-intensive issues, where the computation capability is the main concern. As a result, the peak computing performance TFLOPS and Linpack benchmark are no longer applicable to DI computing, where the huge data processing capability turns out to be the main problem. Recently, plenty of benchmarks have emerged [2,13,5] for DI computing, designed for evaluating the performance of a platform rather than analyzing the DI characteristics of specific applications. Plenty of related factors need to be considered in identifying a problem as data-intensive [18]. The requirements and challenges posed by DI computing vary across different applications. Thus, it is difficult to give a definition covering the full scope of diverse DI applications. Focusing on the large-scale RS data processing, the DI issues are not well defined and analyzed, except for awareness of the huge volumes of RS data.

To properly solve the above issues in the RS domain, we propose DI_{RS}, an empirical model of a data-intensive index for Remote Sensing applications. Our main contribution in this work is a novel model to quantify DI issues in RS data processing with a normalized DI index. RS data processing is normally carried out as a multi-staged workflow that corresponds to the concept of on-the-flow processing [6]. For a thorough analysis, we choose typical algorithms covering the entire processing flow from satellite data acquisition to thematic applications for study. At each processing stage, we specify an empirical DI_{RS} index for each typical algorithm according to experts experience. Here, the possible factors that may influence the DI issues are taken into account for analysis. These factors include data volumes, the data rate, data throughput, complexity of algorithms, and so on. Then, the correlation between these factors and the DI_{RS} index, as well as the contribution of each factor to the total DI_{RS} index, are used for experimentation and quantified. Thereafter, the significant factors are distinguished to model the DI index mathematically. Accordingly, by combining empirical knowledge, experimentation, and quantitative analysis, we construct an empirical (DI_{RS}) model to quantify the DI issues for RS data processing.

The rest of this paper is organized as follows. The Section 2 reviews some related work, and the problem definition is addressed in Section 3. Section 4 presents the analysis of the RS data processing flow as a whole. In Section 5, we go into the detail concerning the construction of the empirical model (DI_{RS}) for quantitatively estimating DI issues in RS applications. The Section 6 discusses the experimental validation and analysis of the DI_{RS} model, and finally the Section 7 summarizes this paper.

2. Related works

The growing amount of datasets is outstripping the current capacity to explore and interpret them [18]. As stated in DOE-sponsored report [1], "we are entering a new era: data-intensive computing". Many organizations and researchers have proposed qualitative definitions of emerging DI issues by emphasizing the huge volumes of data [17,23].

Driven by the widening of application requirements, the definition of data-intensive computing is shifted to a broader realm to focus on the time it takes to reach a solution as a key factor [18]. One definition is as follows: "*computational task where data availability is the rate-limiting factor to producing time-critical solutions*" [18]. Another more promising and comprehensive definition has been put forward by [14]: "*data-intensive computing is managing, analyzing, and understanding data at volumes and rates that push the frontiers of current technologies*" [18,28]. However, both these definition of DI issues have a qualitative focus. Solely depending on these vague definitions will make it almost impossible to classify an application as data-intensive. Thus, some more specific standards for determination are essential. These standards could be estimation indexes or benchmarks. Therefore, the point is that a more quantitative approach to measure and analyze the realm of DI issues turns out to be valuable.

Benchmarks are commonly accepted for performance evaluation. In contrast to compute-intensive problems where Peak Flops, Linpack [8], and TOP500 are accepted as widespread benchmarks, DI issues are much more complicated to measure. The reason for this may be the complexities arising from the extremely large scale of data and the actual geographical distribution characteristics. Recently, a sort of DI benchmarks have emerged where Malstone [2] designed DI computing of data mining in the Cloud, the Sort Benchmark is used for massive data sorting in the Cloud, and Graph 500 [5] uses DI graph computing for estimation. However, all of these benchmarks are used for performance estimation of platforms with given DI requirements of applications, but not for analyzing applications themselves.

Reagan [23] defined DI computing as "*Applications that are I/O bound, and devote the largest fraction of execution time to movement of data*". He also proposed "*Computational Bandwidth[the] number of bytes of data processed per floating-point operation*" as a quantitative way of identify DI problems. Actually, the computational bandwidth describes the data throughput requirement of applications on a platform with a computation capability of certain flops. The mismatch between the evaluated computational bandwidth of applications and the fixed rate of the system (disk bandwidth divided by Peak Flops) would lead to imbalance in the system. From a system perspective, this is an excellent way of quantitatively evaluating DI computing in general. However, this approach does not take application-specific features into account. The truth is that the rising volume of data may increase the complexity of the processing to a certain extent for some applications.

On the other hand, DI computing is a complex issue that varies across different disciplines. A single traditional QoS (Quality of Service) index like response time, data throughput, or number of requests per unit time is no longer applicable for estimating this complex issue. Applications across different disciplines have their own challenges and requirements for DI computing [18]. This is also why we have different DI benchmarks abstracted from various domains. However, for the Earth Observation domain [12], the only consensus on the DI issue is the huge volumes of RS data [25]. There is no clear definition of the DI issue in the RS data applications or benchmark for RS applications, not to mention indexes for quantitative estimation and analysis.

The empirical DI_{RS} index model proposed in this paper aims to address these challenges by offering a quantitative way of estimating DI issues for RS applications. This solution relies on a normalized DI_{RS} index to describe the extent of DI problems underlying RS applications. Based on the comprehensive analysis of the entire flow of RS data processing, we determine the key factors influencing the issue and their relationships. Based on the analysis and empirical knowledge of the existing RS algorithms, we build a mathematical model to describe the DI issue. Relying on this empirical model, we could easily distinguish an RS application as data-intensive and gain insight into the realm of the DI issue.

3. Problem definition

This section demonstrates the primary issues related to the estimation and analysis of the DI computing problems in a large-scale RS data-processing scenario. This problem has two key aspects. The first has to do with the quantitative index chosen to describe DI issues in Remote Sensing applications comprehensively (Section 3.1). The second has to do with how to mathematically model the factors that influence the DI problems most (Section 3.2).

3.1. Data-intensive computing index

RS data generated by a single high-resolution sensor are transmitted at a rate of several gigabits per second. The data exploited for continent-scale forest monitoring add up to several TB [3]. Thus, extremely massive RS data need to be processed. Moreover, the massive RS data exploited for processing may have hundreds of spectral bands and cover a time period of decades. The high dimensionality nature of pixels increases the complexity of the data. Except for the challenges related to the volume and complexity of data, RS data processing has its own application-specific requirements and challenges. These include the intensive irregular I/O patterns [7,21], the multi-staged processing chain [20,22], and the real-time processing requirement [25,26]. Accordingly, a promising index for measuring the DI issue should be a normalized value which comprehensively takes all of the main related requirements into consideration.

The index should also be able to leverage various RS applications involving different data scales. The main reason for this is that in most cases, RS applications evidence different levels of DI computing features.

3.2. Modeling the RS DI issue

The occurrence of the DI computing issue in RS data processing is generally introduced via the joint effort of many related factors. To be specific, these determining factors may involve data volumes and data rates. But not only that, DI issue here probably refers to the data-intensive computing requirement of the RS data processing applications. DI issues of RS applications are also related to the I/O occupation and algorithm complexity of the RS applications that are I/O bound. However, some other factors, including complexity of data, algorithms, and time requirements could also be important [18]. Thus, the problem is how to figure out the key factors affecting the DI issue and the correlation among them. Another issue is how to find an appropriate mathematical model for these key factors. Moreover, the processing as a whole could be treated as a multi-staged train with several individual steps leading to the final output. Thereafter, the modeling should also take multi-stage processing into account.

4. Analysis of the entire data-processing flow

RS data processing has an important role throughout the Earth Observation system. The longest RS data processing flow starts from data acquisition and ends in thematic applications. As showed in Fig. 1, the entire processing flow is commonly segmented into several stages.

1. RS data acquisition, recording, and transmission.
2. Preprocessing.
3. Value-added processing.
4. Information abstraction.
5. Data product releasing.

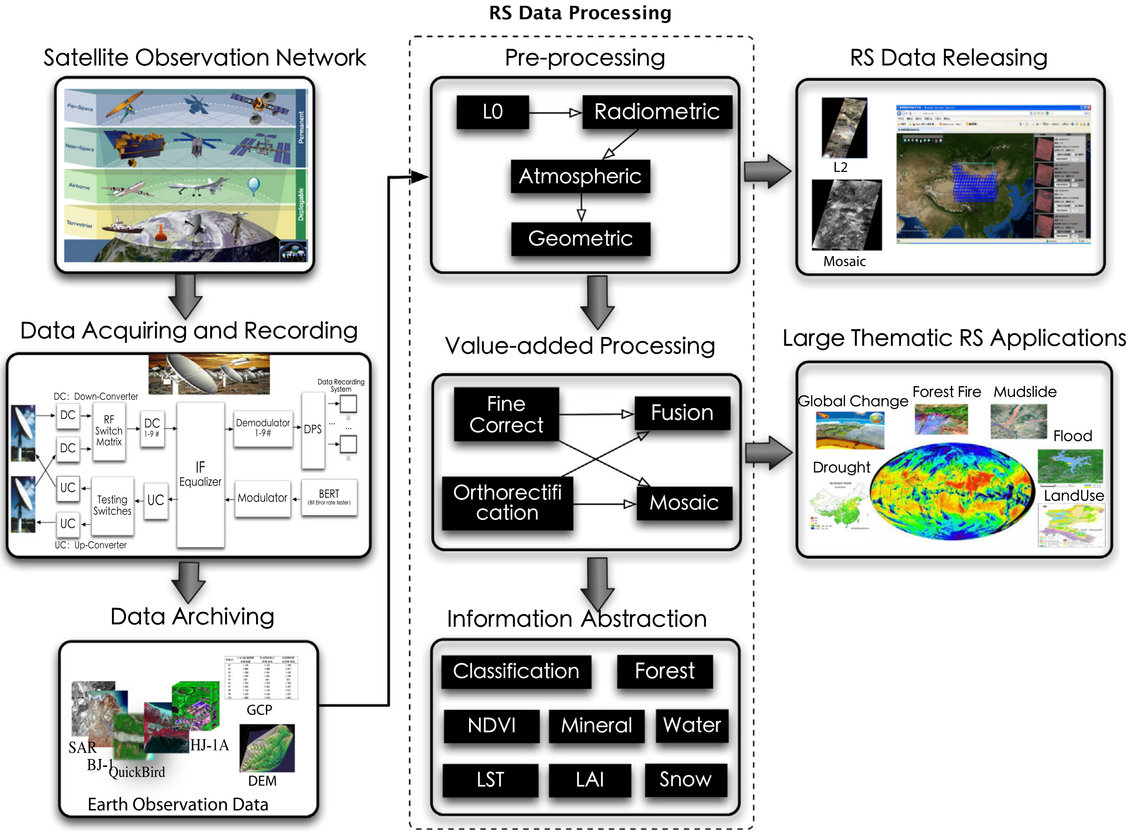


Fig. 1. Analysis of the entire Remote Sensing data-processing flow.

6. Large thematic RS applications.
   Where preprocessing is responsible for some initial processing, including radiometric correction, geometric correction, image enhancement, and so on. Value-added processing aims at processing geo-referenced data, including fine correction, orthorectification, fusion, mosaic, and so on. Meanwhile, the information abstraction stage produces diverse common information like classification, leaf area index (LAI), land surface temperature (LST), normalized difference vegetation index (NDVI), and so on. The thematic application stage refers to RS applications across various disciplines, like drought monitoring, disaster monitoring, and global change.

For deep insight into DI issues, we conducted a thorough analysis throughout the whole RS data processing flow in this section. Several space-borne sensors were chosen as the typical source of the RS data streams. The features of the typical algorithms at each stage of the data processing flow were quantitatively analyzed. Here, we focus on the features that may influence DI computing issues, including data volumes, data rates, data throughput, and data processing speed. By virtue of this analysis, we put forward some discussion of the DI features at each stage.

4.1. Analysis of data acquisition and transmission

The acquisition and transmission of the RS data is the beginning of the Earth Observation flow. We take the actual statistical data from three satellite ground stations of the Chinese Academy of Sciences (CAS) for case study [16,15]. Here, the data volumes and data rates are taken as the main consideration.

4.1.1. Data acquisition

The antennas in the ground stations are responsible for tracking satellite signals and acquiring data from satellites through downlink channels. The aforementioned three geographically distributed ground stations have to deal with dozens of satellites every day. The data rates and the number of download channels vary across different satellites. The data rates of the downlink, the data amount received per day, and the data amount received per year of each satellite are showed in Table 1.

From the statistical data for the ground stations in Table 1, we can determine that the satellite data streams are generated and acquired at a rate of 60 megabits per second (Mbps) to 1.6 gigabits per second (Gbps) if considering high-resolution satellites. The aggregate downlinking bandwidth of the multiple satellites is around 3.7 Gbps and would add up to be more than

Table 1
Data acquisition: data volumes and data rates.

Satellites	Data rates (MB/S)	Data amount (GB/Day)	Data amount (TB/Year)
ZY-02	113.98	115	40.99
ZY-03	304	334	119.05
HJ-1A	120	114	40.63
HJ-1B	60	57	20.32
HJ-1C	320	187.5	66.83
ZY-3	900.00	494.38	96.80
ZY-02C	320.00	175.78	34.42
SJ-9A	190.00	104.37	20.44
LandSat8	440.00	241.70	47.33
RadaSat2	105.00	57.68	11.29
RadaSat-1	105.00	57.68	11.29
SPOT-5	100.00	54.93	10.76
LANDSAT5	85.00	28.02	9.99
SPOT-4	50.00	10.99	3.92
RADARSAT-1	105.00	34.61	12.34
ENVISAT	100.00	32.96	11.75
IRS-P6	210.00	46.14	16.45
Total	3712.98	2089.06	574.6

10 Gbps. The volume of data acquired from a single satellite is about more than 500 GB every day. The total volume of data acquired by three ground stations connecting the data center sums up to about 2 terabytes per day, and exceed 0.5 petabytes per year. The total amount of data acquired by ground stations reaches about 6 TB per day and exceeds 1 petabyte per year. Not only that, but with the emergence of satellites with even higher spatial and temporal resolution, the data volumes will rise exponentially.

4.1.2. Data transmission

The acquired data streams are simultaneously transferred to the satellite data center. In order to match the actual aggregated data rate of the satellite data streams, the bandwidth requirement of the network for data transmitting should be dozens of gigabits per second. However, in actuality, the bandwidth of the recent dedicated network over the Internet connecting the data center and geographically dispersed ground stations is about only 155 Mbps. In other words, the rate of data transmission is lower than the rate of data acquisition. This situation makes real-time data transmission unattainable.

4.2. The analysis of data preprocessing

The preprocessing of RS data typically involves some initial processing of raw satellite RS image data and generates corresponding levels of data products.

4.2.1. L0 processing

Level 0 processing, or L0 for short, conducts frame synchronization, descrambling, decompression, and scene framing of the original RS data streams. Except for the multi-dimensional image data, the L0 data products also contain some auxiliary data, including GPS, orbit parameters, navigation data, and timing data.

4.2.2. Radiometric correction

Radiometric correction is responsible for calibrating the radiometric distortion. It involves the correction of sun elevation, earth-sun distance, and haze compensation. The processing of these corrections will produce L1 data products.

4.2.3. Geometric correction

This processing aims at correcting the geometric distortions caused by variations in orbit, sensor platform, and earth rotation. Following data geo-referencing outcomes, L2 data are produced.

In the analysis of the preprocessing stage, we included several typical satellites in the case study, such as BJ-1, and LANDSAT7. The data volume of a single standard scene of L0, L1, and L2 RS data products of these satellites are listed in Table 2. From the data in this table, we can see that the volume of the RS data increases during the data preprocessing flow. The main reason for this is that geometric correction will enlarge the data during the geometric mapping, similar to image rotation. In addition, some of the raw data are compressed. For example, the raw data for HJ-1C [29] are compressed at a rate of 8–3. Thus, the L0 processing of HJ-1C would be bound to triple the data.

The data processing speed QoS of each processing algorithm L0, radiometric correction, and geometric correction are shown in Table 3.

These statistical data are gathered from the actual running satellite data processing system. The data processing speed when processing a single scene of satellite RS image is given. For a reasonable comparison, the L0, geometric correction,

Table 2
Data volumes of a single scene of L0, L1, and L2 products.

Satellites (Sensor)	L0 (MB)	L1 (MB)	L2 (MB)
BJ-1(MS)	228.3	228.3	333.32
BJ-1(PAN)	98.9	98.9	150.33
ZY-02C(PAN)	120.87	120.87	168
ZY-02C(HRI)	270.82	270.82	393
SPOT-5(PAN)	572	572	45.96
SPOT-5(MS)	144	144	252.21
LANDSAT7(ETM)	337	337	532
LANDSAT8(UTM)	481	481	781

Table 3
Speed of processing L0, L1, L2 data products.

Satellites (Sensor)	L0: decomposition (MB/s)	L1: radiometric (MB/s)	L2: geometric (MB/s)
BJ-1(MS)	2.65	0.12	0.25
BJ-1(PAN)	2.71	0.14	0.76
ZY-02-C(PAN)	3.78	2.24	1.21
ZY-02-C(HRI)	9.03	1.79	0.87
SPOT-5(MS)	2.45	0.41	0.21
SPOT-5(PAN)	3.56	0.62	0.75
LANDSAT7(ETM)	4.9	0.38	0.31
LANDSAT8(UTM)	4.3	0.31	0.28

and radiometric processing are all executed on a single processor, since some of this processing is implemented as MPI-enabled parallel programs. For different sensors of the satellites, the data processing algorithms and steps differ from one another. From the statistical data, we could say that the data preprocessing is rather time-consuming. Comparatively, the speed of preprocessing is far lower than the rate of data acquisition.

4.3. Analysis of value-added processing

Value-added processing is responsible for some further processing of RS data, including fine correction, orthorectification, fusion, and mosaicking. Fine correction is used for further correction of the geometrical distortion using reference data, while orthorectification using DEM (Digital Elevation Model) data is employed to correct the parallax resulting from the undulating terrain. Mosaicking usually stitches a collection of RS image scenes to form a seamless continuous view of a large area. It features a complex processing chain, enormous computation, and massive amounts of data. Taking ZY-02C and LANDSAT7 as a case study, we demonstrate the data amount and the data processing speed of a single process in Tables 4 and 5.

The data volume for a regional mosaic of Landsat (ETM: Enhanced Thematic Mapper) reaches 7.48 GB, while the volume of a nationwide mosaic adds up to more than 250 GB. The amount of data is far larger than the fine correction and orthorectification. Compare to fine correction or orthorectification, the processing speed of mosaicking is relatively very poor. Generating a regional mosaic takes about 1 h. The situation will be even worse in case of advances in the sensors.

Table 4
Data volume of value-added processing.

Satellites (Sensor)	Fine-rectification	Ortho-rectification	Regional mosaic (GB)	
	1 (MB)	1 Scene (MB)	15 Scenes	512 Scenes
LANDSAT7	511.52	515.3	7.48	255.5
ZY-02C(PAN)	201.61	206.65	N/A	N/A
ZY-02C (HRI)	467.3	475.15	N/A	N/A

Table 5
Speed of value-added processing.

Satellites	Fine-rectification	Ortho-rectification	Mosaic (GB/s)
MB/s	1 Scene	1 Scene	15 Scenes
LANDSAT7	0.43	0.36	0.023
ZY-02C(PAN)	0.89	0.95	N/A
ZY-02C(HRI)	1.04	0.9	N/A

Table 6
Data volumes of single scene of L0, L1, and L2 products.

Algorithm	Data amount (MB)	Time (s)	Speed (MB/s)
K-mean (115 km * 130 km)	380	320.40	1.19
K-SNN (15 km * 15 km)	27.5	832.00	0.05
Nation-wide aerosol retrieval	4034.5	276540.00	0.01
Oil spilling	130	600	0.22

4.4. The analysis of information extraction and thematic applications

Information extraction involves classification, feature extraction, and quantitative retrieval. To be specific, this processing stage produces common Remote Sensing information products, including NDVI, LST, LAI, snow-covered area, and so on. Some of these algorithms are of high complexity, especially quantitative retrieval, which always involves nonlinear equations problems.

Thematic applications include some regional to global scale problems covering various disciplines, including agriculture, forestry, mining, and as resource and environmental monitoring. To specify, these applications normally exploit large regions with multi-temporal RS data as input for processing. The time period may be a month, a quarter, or a year, depending on the applications.

In these two processing stages, we choose K-mean and k-SNN classification, as well as nationwide aerosol retrieval and oil spilling as case studies. The comparative analysis is listed in Table 6.

As shown in Table 6, the processing of these algorithms are extremely time-consuming. In particular, nationwide aerosol retrieval using 40 GB MODIS RS data, would take about 76 h. Overall, the data-processing speeds of these algorithms are relatively poor compared to the algorithms in other processing stages.

5. Empirical model of DI_{RS} for RS applications

In this section, we demonstrate DI_{RS}, an empirical model of the DI index, to estimate the DI issues normally occurring in RS applications. This employs DI_{RS}, a novel normalized index, to easily quantify the degree of DI computing. Relying on this model, we could intuitively classify an RS application as data-intensive and leverage the degree of DI computing as well. To resolve the issues discussed in Section 3.1, we adopt a comprehensive index to estimate the DI issue instead of a simple QoS index solely. All of the major related factors influencing the DI issue are taken into account in the index. That is because the DI issue is a complex problem generally caused by the joint efforts of several factors. Except for the characteristics of the RS data, we also take the features of the various RS application as a considerable factor. The main reason for that is this index needs to be capable of distinguishing the different levels of DI requirements across various RS applications.

To resolve the issues mentioned in Section 3.2, we combine the empirical estimation of the DI problem, as well as experimental and quantitative analysis of key factors together in building the empirical DI_{RS} model. The main concept behind this approach is demonstrated in Fig. 2.

1. First, we investigated the factors that may affect or give rise to the DI issues through some rough qualitative analysis. Concerning the characteristics of the RS data and the algorithms of RS applications, the factors under consideration should at least include data volume and the complexity of the algorithm.
2. Second, we selected typical algorithms from the entire RS data-processing flow for experimental analysis of the DI problem. The DI analysis includes the analysis of computational complexity (C), data volume (D), and the experimental analysis of the performance indexes of data throughput and Flops/IO rate. Wherein, the Flops/IO here refers to the ratio of computing time and the time used for data I/O. Based on these analyses, we draw the corresponding curves of these performance indexes, which are also treated as the potential key factors.
3. Third, we conducted empirical estimation of the DI issue according to the experimental throughput curve of the algorithms. Based on the actual empirical knowledge, the normalized value was offered to indicate the degree of DI for each algorithm. Then we could get the curve of the empirical DI estimates (DI_{RS}).
4. Finally, we selected the corresponding fitting functions to describe the quantitative relationship between the DI_{RS} estimates and each potential key factor (D, C, T). Based on the quantitative relationship, we built an empirical model (DI_{RS}) to describe the DI index of RS applications.

5.1. Comparative analysis of key factors throughout the processing flow

Concerning the characteristics of both RS data and the algorithms of RS applications, the factors under consideration should at least include the data volume and complexity of the algorithm. On the other hand, with the increasing requirements of the time-critical applications, the requirements of real-time processing aggregate the problems of DI computing. In this way, we also considered the time requirement as a potential factor in the DI problem. Consequently, we identified the data volume (D), computational complexity (C), and time requirement (T) as potential factors influencing DI problem.

Please cite this article in press as: Y. Ma et al., Towards building a data-intensive index for big data computing – A case study of Remote Sensing data processing, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.10.006

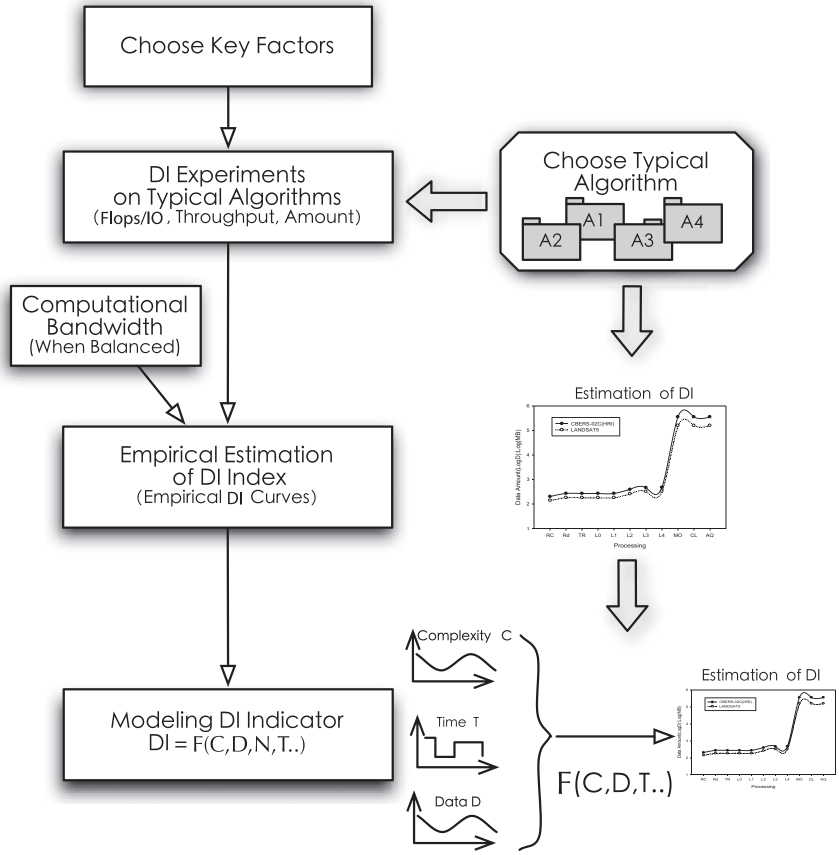


Fig. 2. Main concept in constructing the DI_{RS} model.

Although a single performance index of QoS could not fully describe the DI problems, these indexes did reflect some aspects of DI. Based on this consideration, we choose the data throughput index and I/O occupation [23] to roughly reflect the trends of the DI issue empirically.

To fully elucidate the DI problem, we conducted experiments on the selected typical processing and algorithms throughout the entire RS data-processing flow. These typical processes included data receiving (RC), data recording (RD), and data transmitting (TR) at the first stage; zero-level processing (L0), radiometric correction (L1), and geometric correction (L2) at the preprocessing stage; fine correction (L3), orthorectification (L4), and mosaicking (MO) at the value-added processing stage; and classification (CL), aerosol retrieval (AR), and oil spilling (OS) at the information extraction and thematic application stages. In addition, the Landsat5 and ZY-02C were taken for case study as sources of satellite data streams.

5.1.1. Comparative analysis of data volume

Throughout the processing flow, we take the satellite Remote Sensing data exploited for a single normal processing. Ordinarily, except for the mosaicking (MO), aerosol retrieval (AR) and oil spilling (OS), all the algorithms will process one single scene at a time. For the sake of comparability, the MO, AR, and OS algorithm exploits Landsat5 TM data for processing, which has nationwide coverage and whose volume adds up to more than 350 GB.

From the data volume of each process shown in Fig. 3, we can observe that the data volume increases along the RS data processing flow. At the preprocessing stage, the data volume increased by 45% compared to the original raw data, while the value-added processing stage undergoes growth of about 20% compared to the preprocessing stage. In particular, when the processing comes to the information extraction and thematic application stage, the data volume exhibits a sharp increase. This is because the mosaicking (MO) and AR tend to exploit the large region covering an amount of data adding up to more than 350 GB. The volume of the data is the far greater than that of previous stages.

5.1.2. Comparative analysis of the computational complexity of algorithms

The computational complexity of each algorithm along the whole processing flow is illustrated in Table 7, where '*p*' stands for the numbers of the principal components, '*b*' stands for the number of spectral bands, '*r*' stands for the number of iterations, and '*m*' stands for the number of classifications.

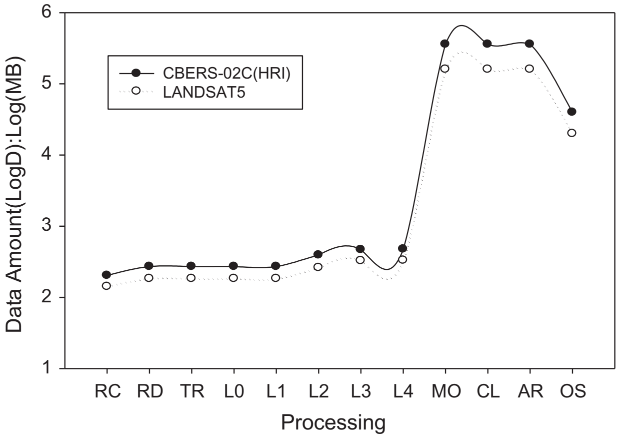


Fig. 3. Comparative analysis of data volume throughout the entire flow.

Table 7
Computational complexity of algorithms.

Stage	Algorithm	Complexity	Parallelism
Preprocessing	Radiometric	$O(n)$	Excellent
	Geometric	$O(n^2)$	Excellent
	Registration	$O(n^2 log(n))$	Excellent
Value-added processing	K–T trans	$O(p^2 bn^2)$	Good
	Convolution	$O(k^2 n^2)$	Good
	Mosaicking	$O(n^3)$	Normal
Information extraction	K-mean	$O(rbmn^2)$	Normal
	Bayers	$O(bmn^2)$	good
	BP	$O(n^2 log(n))$	good

From the curve shown in Fig. 4, the computational complexity of the beginning of the processing flow is relatively low, such as the beginning of the preprocessing stage. By contrast, the L2 of the preprocessing stage, the value-added processing, information extraction, and applications are characterized by relatively higher computational complexity.

5.2. Estimating the data-intensive issue with the empirical DI_{RS} indicator

According to Reagan's point of view [23], the fraction of execution time devoted to data movement is a major concern for identifying the DI computing problems. Thus, we carry out experiments on the I/O time of the algorithms and analyze the

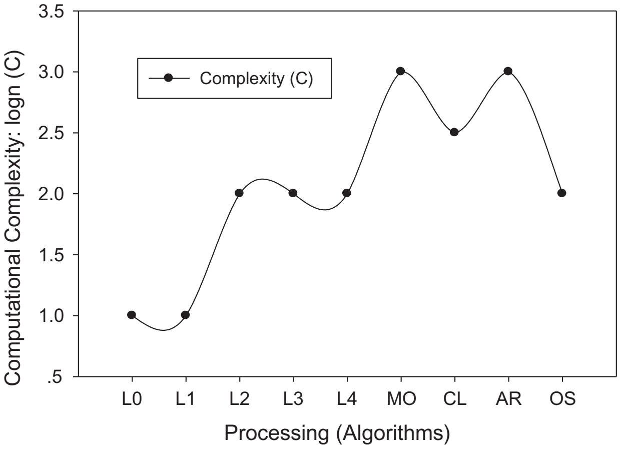


Fig. 4. Comparative analysis of computational complexity of algorithms.

occupation of I/O in the total execution time. The algorithms are experimented with same volume of RS data. The curves of the I/O occupation are demonstrated in Fig. 5.

As shown in Fig. 5 we can see that the algorithms with higher computational complexity occupy a much smaller fraction of I/O time. The data I/O occupation of the MO is about 20–40%, while for other algorithms, the I/O takes charge of more than half of the total execution time. On the other hand, with the data scale increased from 500 MB to 22 GB, the I/O occupation also rises gradually. Accordingly, the curves of I/O occupation do reflect the DI computing issue to a certain extent.

The data throughput rate is an important performance indicator for QoS. For an analysis of the DI issue from a performance point of view, we also carried out the data throughput rate experiments on algorithms across the whole RS data processing flow. As illustrated in Fig. 6, the data throughput rate of the algorithms decreases dramatically along the whole processing flow. The satellite data stream was generated and acquired at a rate of 60–1080 GB, while the speed of data preprocessing is normally less than 2 MB/s, and the speed of the value-added and other applications is even worse than 1 MB/s. By contrast, the data throughput rates of these processing algorithms just reach 1–30% of the data rates of the satellite downlink. Obviously, the data-processing speed is far greater than the rate at which data are acquired. The main reason for the dramatic decline of the data throughput rate must be the increase in the computational complexity of algorithms and the amount of data.

By virtue of the analysis of data throughput rate and I/O occupation, we explored the empirical DI_{RS} estimation of DI issues in typical RS data processing algorithms. Wherein, the DI_{RS} here is namely a rough quantitative estimation based on the empirical knowledge of experts in Remote Sensing domain. DI_{RS} not only indicates the volumes and rates of data that processed, but also related to the I/O occupation and algorithm complexity of the RS applications that are I/O bound. Accordingly, based on the rich practical experiences of experts, the RS applications with larger amount of data, lager fraction of data movements as well as relatively higher complexity of algorithms are probably recognized as issues with higher degree of DI computing. Following this way, these RS applications with higher degree of DI computing issue would be assigned with a

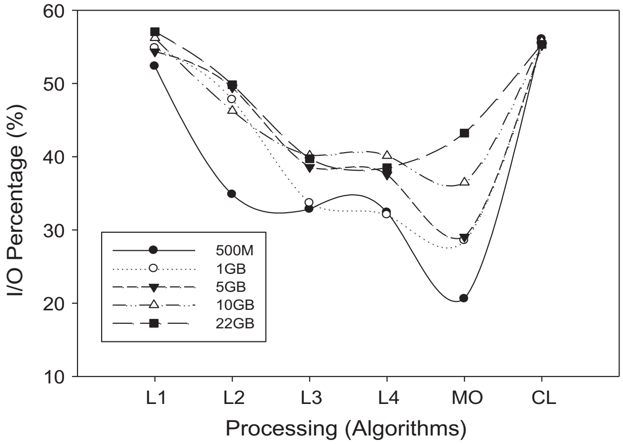


Fig. 5. Comparative analysis of I/O occupation.

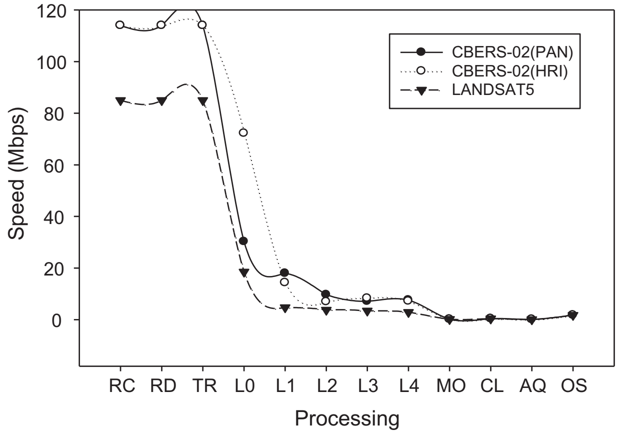


Fig. 6. Comparative analysis of the data throughput rate.

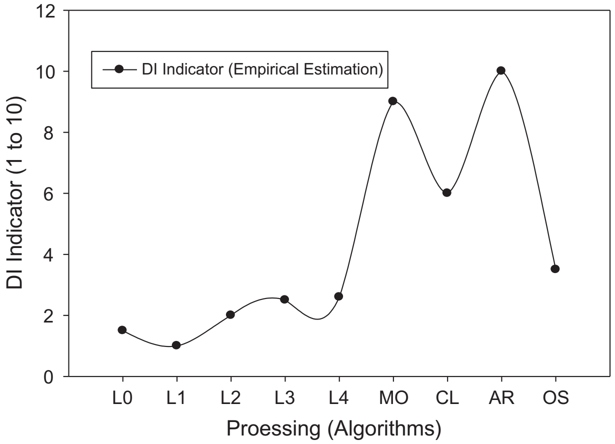


Fig. 7. Empirical estimation of the DI_{RS} indicator.

relatively higher DI_{RS} estimation. Empirically, the experts could give the approximate index values to distinguish the degree of DI computing for different RS applications. The empirical DI_{RS} estimation of the typical algorithms along the processing flow is drawn as Fig. 7. Here, the absolute value of the empirical DI_{RS} index is not critical, while the partial order of these values is more concerned with the capability to distinguish among various algorithms. For instance, the nationwide MO and AR are assigned with relatively higher empirical estimation of the DI_{RS} index.

5.3. Analysis of the correlations between factors and DI_{RS}

The DI_{RS} index proposed in this paper probably indicates the degree of the DI computing in the RS data processing. It is a comprehensive index that combines several related factors rather than a single performance index. Based on the above experimental analysis of the selected factors, together with the empirical estimation of the DI_{RS} index, we have probed into the correlations between these factors and the DI_{RS} index. Comparing the curve of the data volume (D) in Fig. 3 with the empirical estimation of DI_{RS} in Fig. 7, we could determine that these two curves had almost the same trend of rising along the processing flow. That is to say that the data volume factor positively affected the DI_{RS} index. This conception fits the claim that the enormous amount of data is the main cause of the DI issues. By comparing the curve of computational complexity (C) in Fig. 4 with the DI_{RS} curve in Fig. 7, we can see that the algorithms with higher computational complexity also had a higher estimated value of the DI_{RS} index. Accordingly, we also took the computational complexity (C) as a positive factor, as RS data-processing applications [25] are normally regarded as both computing- and data-intensive. The computation of algorithms greatly depends on the RS data. Thus, the increase in data would bring in more computational increase for the algorithms with higher computational complexity. In this way, we developed the following Eq. (1)

$$DI_{RS} \propto C, DI_{RS} \propto D \tag{1}$$

Reagan [23] suggested using the computational width (CW) to identify DI problems. The CW index refers to the bytes of data to be processed per floating-point operation. To be more specific, we can infer the equation for the calculation of CW as Eq. (2). The Flops may use the peak flops (TFLOPS) or the Linpack value, where 'D' represents the data volume, 'T' is the time requirement of the user or applications, and TFLOPS is the peak float point operations per second:

$$CW = \frac{D}{T * TFLOPS} \tag{2}$$

According to Eq. (2), we can determine that the time requirement of applications negatively influences the CW index, and even DI problems. With the increasing demand for real-time processing capability by time-critical RS applications, the time requirement is also an important factor that should be taken into account for the analysis of the DI problems of RS applications. Thereafter, we could also model out the correlation between factor time T and DI_{RS} index as Eq. (3):

$$DI_{RS} \propto \frac{1}{T}, \tag{3}$$

It is reasonable that an applications demand for a short time requirement would inevitably lead to the requirement of a high data throughput rate, as this kind of requirement also processing extensive amounts of RS data in a very short time. Eventually, this requirement would also be followed by intensive data I/O and processing, which would greatly challenge the existing system.

5.4. Empirical DI_{RS} model for RS applications

The building DI_{RS} model in relates to how to find a proper mathematical function F(D,T,C) to describe the aforementioned correlation between the factors (D, C, T) and DI_{RS}. Based on the analyses above, we propose the definition of the DI_{RS} shown in Eq. (4):

$$DI_{RS} = \begin{cases} 0 & \textbf{if } \frac{10^{DC}}{60^T} < 1 \\ \log \frac{10^{DC}}{60^T} & \textbf{if } \frac{10^{DC}}{60^T} > 1 \end{cases} \tag{4}$$

In the DI_{RS} model, the factors of D, C and T of the DI_{RS} model are magnitude values rather than the exact absolute value. "10" represents the magnitude of the data amount, like one gigabytes equals to 10^3 MB, while "60" represents the magnitude of the time, like one minutes equals to 60 s. 10^D reflects the max megabytes of data throughput during the computation procedure at a certain extent. However, this is not an actual data amount. Meanwhile, 60^T represents the time requirement of the applications.

It is known that the computational complexity 'C' is closely related to the computation. Actually, 'C' could be read as the magnitude of the computation. The computation 'Comp' could be calculated as D^C. Accordingly, 10^D refers to the estimation of data volume. While, 10^{DC} is used in this model to describe the relation between data volume 'D' and algorithm complexity 'C'. 10^{DC} mostly reflects the total computation introduced by RS data at volume of 'D' with the algorithm complexity 'C'. Where 'C' represents the magnitude of problem scale that also means volumes data processed by RS applications. Thereafter, the DI_{RS} index to some extent reflects the expectation of data throughput per unit time for applications. As we can see, DI_{RS} is a comprehensive index. The data amount here is not the only determining factor. The applications dealing with huge amounts of data would not necessarily introduce DI computing when the time requirement 'T' is large enough and complexity 'C' is small enough.

The mapping relation between the data volume, computational complexity and time requirement is: 10^D refers to the estimation of data volume; the 10^{DC} also means $(10^D)^C$ is the computational requirements with the algorithm complexity C, where C represents the magnitude of problem scale (also means data volume here for RS applications).

5.4.1. Representation of data volume 'D'

Normally, RS applications are multi-staged. This means that the whole processing of the algorithm is a processing chain consisting of several steps. Actually, the output data for each step are passed through to the next processing step as input. Therefore, during the general RS data processing procedure, the data shown in Fig. 8 may involve input data (D_{in}), intermediate data (D_m), and the output result data (D_{out}). As is depicted in Fig. 3 the data amount undergoes a gradual increase along the whole RS data processing flow. This probably means that the output data D_{out} of the processing would be bigger than the input data D_{in}. In this way, we define the data volume 'D' of the RS applications as Eq. (5):

$$D = \max\{D_{in}, D_{m1}, D_{m2}, \ldots, D_{out}\} \tag{5}$$

RS applications usually exploit multi-dimensional RS data covering different scales of geographical regions and different time spans. The RS data covering different scale of regions are commonly of different data volumes. The data volume of a regional scale could be hundreds of megabytes to several gigabytes, nationwide data could be hundreds of gigabytes, and data at the continental scale could even reaches TB. In this situation, the data volume 'D' always spans a large range. Therefore, in order to form a normalized value for DI_{RS} index, we divide the range of 'D' into 10 sub-ranges and quantify these with values in the range of [1,10]. The mapping of D to the data volume range is listed in Table 8.

5.4.2. Representation of computational complexity 'C'

The factor for computational complexity 'C', which determines the magnitude of the computation, is normally presented in O(n) format, as in $O(n), O(log^n),$ and $O(n^2)$. In order consider 'C' as a factor of the DI_{RS} index model, it is necessary to quantify 'C'. In this paper, we grade the computational complexity 'C' and quantify the grades. The quantization of the factor 'C' is listed in Table 9.

5.4.3. Representation of time requirement 'T'

Most of the RS application have time requirements, especially for some time-sensible applications, like hazard tracking and assessment. The demands for real-time or near real-time processing always pose the challenge of DI computing problems. Therefore, the time requirement is also a very important factor in the DI_{RS} index.

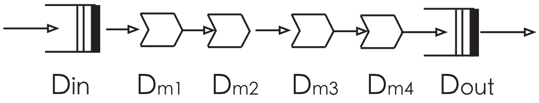


Fig. 8. The representation of 'D'.

Table 8
Quantization of data volume 'D'.

Data volume	'D'
<10 MB	1
10–100 MB	2
100 MB–1 GB	3
1–10 GB	4
10–50 GB	4.5
50–100 GB	5
100–500 GB	5.5
500 GB–1 TB	6
1–10 TB	7
10–100 TB	8
100 TB–1 PB	9
>1 PB	10

Table 9
Quantization of computational complexity 'C'.

Computational complexity	'C'
$O(n)$	1
$O(n^2)$	2
$O(n^2 log(n))$	2.5
$O(k^2 n^2)$	2.2
$O(n^3)$	3
$O(n^3) \sim O(n^4)$	4

'T' refers to the requirement of time to solution. As mentioned in the "Analysis of the Correlations between Factors and DI_{RS}", this is a negative factor of DI_{RS}. The higher the 'T', the larger the time span is, which will naturally lead to the low degree of the DI computing, and vice versa. This is because the applications could take their time for processing in this situation. We also offered a quantization of the time requirement listed in Table 10.

5.4.4. Estimating DI problems with DI_{RS} index

Based on the proposed DI_{RS} model, we could easily give the empirical estimation of DI problems in the typical algorithms along the processing flow. The estimation is listed in Table 11 and illustrated in Fig. 9. The factors 'D' and 'C' are respectively listed in Figs. 3 and 4. We suppose that the time requirement 'T' of the oil spilling (OS), MO, classification (CL), and AR are 3, since these algorithms have to process regional or even nationwide regions. For other algorithms, the 'T' factor would be 2. From Fig. 9, we then see that the nationwide, large-scale mosaicking contributes to a DI_{RS} of 11.2 within a couple of hours,

Table 10
Quantization of time requirement 'T'.

Time requirement	'T'
Second	1
Minute	2
Hour	3
Day	3.5
Week	4
> Month	5

Table 11
Empirical estimation of DI_{RS} through the entire RS data processing flow.

Algorithms	'C'	'D'	'T'	DI_{RS}
L1	1	3	2	0
L2	2	3	2	2.4
L3	2	3	2	2.4
L4	2	3	2	2.4
MO	5.5	5.5	3	11.2
CL	5.5	3	3	8.4
AR	5.5	3	3	11.2
OS	4.5	3	3	3.7

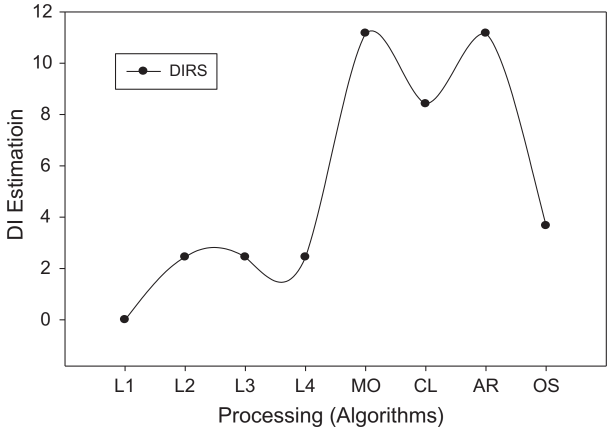


Fig. 9. Empirical estimation of DI_{RS} through the entire RS data processing flow.

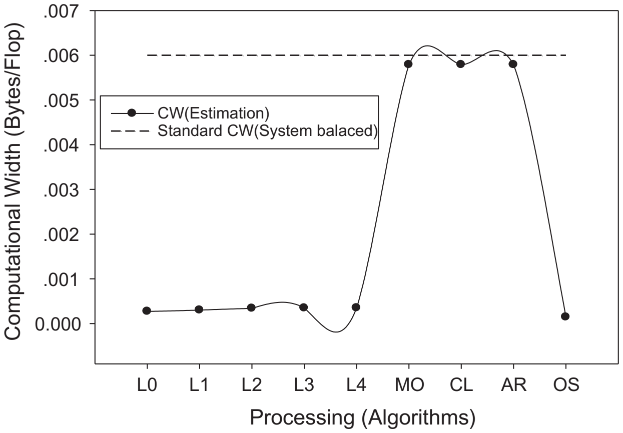


Fig. 10. CW estimation of algorithms in RS data-processing flow.

while a normal preprocessing (L2–L4) of a single scene of RS image data results in a DI_{RS} value of 2.4. Actually, the maximal DI_{RS} value in this model is about 50 for the max ‘D’ and ‘C’, together with the minimal ‘T’. So the range of the DI_{RS} index value is [0, 50]. The DI_{RS} based estimation of these typical algorithms is consistent with the empirical estimation of domain experts.

With the DI_{RS} model, we were able to distinguish different degrees of DI problems across various applications. The point is that in the range of [0, 50], the DI_{RS} value should be identified as really data-intensive. According to the CW approach of Reagan [23] illustrated in Eq. (2), we could estimate the CW index of the RS algorithms along the processing flow. Suppose that we implement these algorithms on one processor equipped with two cores of 3.0 GHZ. This node could offer a TFLOPS of about 24 Gflops. We also suppose that the bandwidth of the local disk would be about 100 megabytes per second. Accordingly, we could give the ‘CW’ estimation of all the algorithm in Fig. 10. As depicted in Fig. 10, for a balanced application in this situation (one processor), the ‘CW’ value should match the rate of 0.0061 (‘balanced CW’) bytes of data per float point operation. This means that the applications transmit 1 byte of data every 163.84 float point operations. Here the algorithms MO and AR have reached a ‘CW’ index value of 0.58, which is close to the ‘balanced CW’ value here. However, when the value of ‘CW’ outstrips the value of ‘balanced CW,’ DI computing challenge would probably occur. So, we choose the DI_{RS} index value of the MO algorithm of 11 as a boundary to identify DI problems. That is to say, when the DI_{RS} index value exceeds 11, we would consider this application to be really data-intensive.

6. Experimental validation and analysis of the DI_{RS} index

The empirical DI_{RS} index presented above offers an easy but promising solution for estimating and identifying DI problems in RS data processing. The experimental validation and analysis of the DI_{RS} model is carried out on a multi-core cluster with 10 multi-core nodes connected by a 20 GB Infiniband using the RDMA (Remote Direct Memory Access) protocol. Each node is a blade server with dual Intel (R) Quad core CPU (3.0 GHz) and 8 GB memory. The operating system is Cent OS5.0, the

C++ compiler is the Intel C/C++ compiler with optimizing level O3, and the MPI implementation is Intel MPI. In this experiment, we choose some more algorithms scattered across the whole processing flow for validation. These algorithms include image mirror (FL), Lucy–Richardson deblurring (LR), Laplace fusion (LF), gradient fusion (GF), pole exponential transformation (LogP or LP), MO, minimum distance classification (MD), k-mean classification (KM), desert extraction (DE), water extraction (NDWI or CW), and normalized difference vegetation index (NDVI or ND).

6.1. Evaluating computational complexity of algorithms – 'C'

For estimation of the DI computing problems with the DI_{RS} model, we first have to analysis the computational complexity 'C' of the RS algorithms. These RS algorithms generally have different characteristics of computing. Take algorithm NDVI (ND) for example, where relevant pixels of multiple spectral bands should all be involved in the calculation of the NDVI index value of this pixel. For each pixel, the kernel calculation of the NDVI is as in Eq. (6), where NIR and R represent the spectral reflectance of the pixels, respectively, in spectral band NIR and R. In NDVI processing, each pixel of the relevant spectral bands performs an arithmetic calculation. Thus, we can infer that the computational complexity of NDVI is about O(n).

$$NDVI = \frac{NIR - R}{NIR + R} \tag{6}$$

On the other hand, for the K-mean classification algorithm with k classifiers, this involves continuous iteration of the kernel classification functions until classification converges. The kernel classification functions is shown in Eq. (7); here, d_{ij} is the distance from the mean value point to the center of this classification. Thus, the computational complexity of k-mean should probably be $O(rbmn^2)$, where, 'r' is the time of iterations, 'b' is the number of spectral bands, and 'm' is the number of classifiers.

$$d_{ij} = \sqrt[\lambda]{\sum_{k=1}^{n} |x_{ik} - x_{jk}|^{\lambda}} \tag{7}$$

Consequently, the factors 'C' for each algorithm are listed in Table 12.

6.2. DI_{RS} estimating of various RS applications

Given the computational complexity of the algorithms, we empirically estimate the DI_{RS} indictor value of all the selected algorithms with the DI_{RS} model. The DI_{RS} estimation experiments are conducted with different volumes of data (500 MB, 10 GB, 100 GB, 500 GB) and different time requirements (several seconds, several minutes). As demonstrated in Fig. 11, with the increase of data volume 'D', the DI_{RS} values of the algorithms all go up gradually. This means that the increase of the data amount truly influences the DI problems positively. Some of the DI_{RS} values have even exceeded the boundary value of 11. This indicates that the algorithms under these requirements of rapid processing data 'D' in time 'T' would inevitably result in DI issues.

6.2.1. Experimental analysis of the data throughput rate

We carry out comparative experiment of data throughput on the above algorithms. The experimental results are illustrated in Fig. 12. In this experiment, 10 processors are employed for implementing these MPI-enabled RS algorithms. From the curves in Fig. 12, we can infer that the increase of data does not greatly affect the total data throughput rate. The algorithms with higher computational complexity 'C' seemed to perform poorly in the data throughput rate experiment. This may have been because of the large amount of data waiting for previous data to be processed through the complex, time-consuming operations. In contrast, for the algorithms with lower 'C', the data throughput rates are relatively high, since the data could be processed rapidly for the easier calculation. On the other hand, as shown in Fig. 12 and Table 12, the algorithms with higher 'C' in this experiment also exhibited higher demand for DI computing (DI_{RS}). However, these algorithms with higher DI_{RS} index values showed poor performance in terms of the data throughput rate. As a result, we could infer that with the requirements of a much higher degree of DI computing (high DI_{RS}), these algorithms would suffered greatly due to the existing processing systems.

Table 12
Computational complexity of algorithms 'C'.

FL	LR	LP	LF	GF	MO	MD	KM	DE	CW	ND
1.1	1.2	1.8	1.6	1.5	3	2.3	2.2	2.1	2.4	1.5

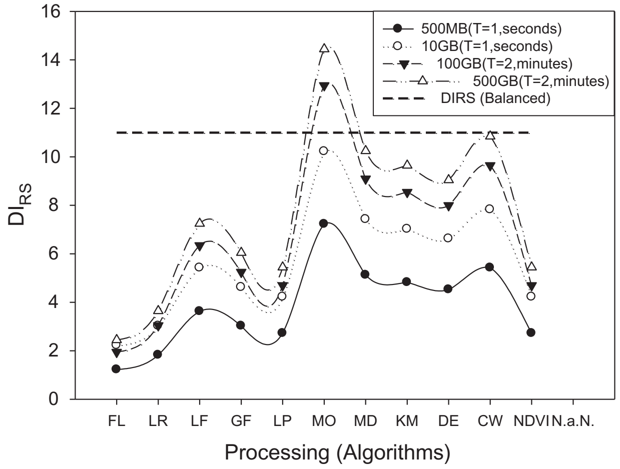


Fig. 11. DI_{RS} estimation of various RS applications with different ‘D’ and ‘T’ values.

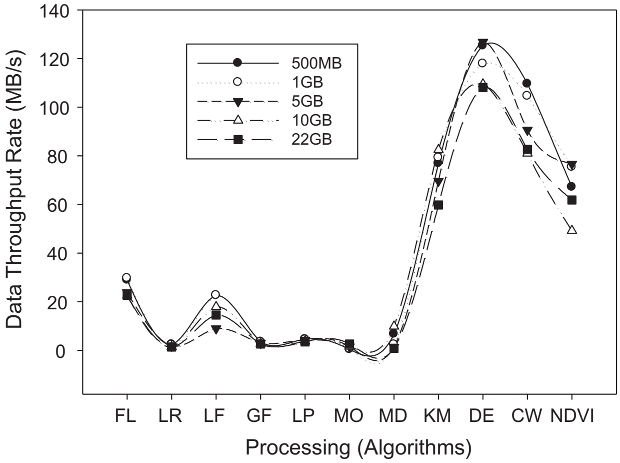
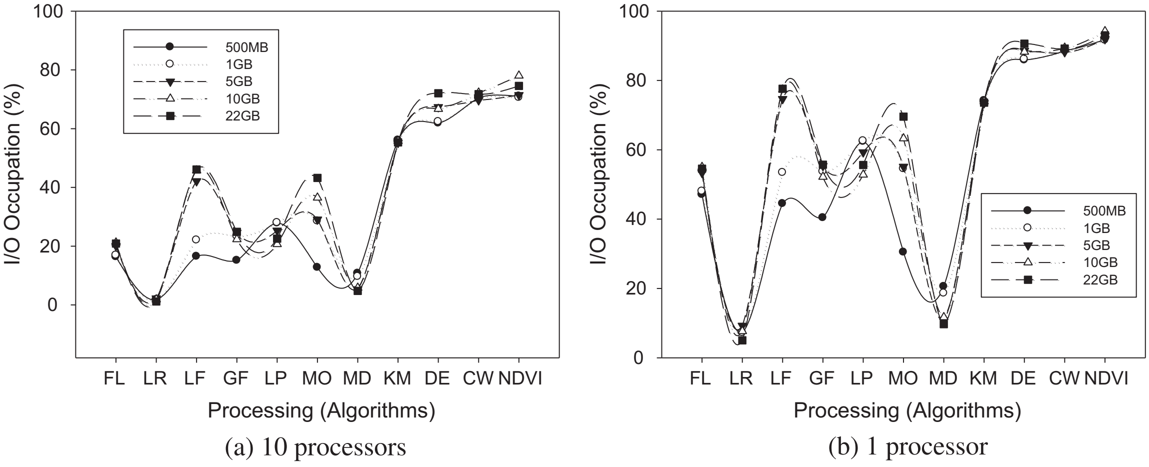


Fig. 12. Data throughput rate of various RS applications.



(a) 10 processors

(b) 1 processor

Fig. 13. I/O occupation of various RS applications.

Please cite this article in press as: Y. Ma et al., Towards building a data-intensive index for big data computing – A case study of Remote Sensing data processing, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.10.006

6.3. Experimental analysis of I/O consumption

We conducted I/O consumption experiments on all of the algorithms, and calculated the I/O occupation of the total execution time. The I/O occupation curves are shown in Fig. 13. The algorithms with higher 'DI_{RS}' seemed to have a higher I/O occupation for the total execution time in the experiment. In contrast, some of the algorithms with lower 'C' also had a high actual I/O occupation. Compared with the algorithms implemented with a single processor (Fig. 13a), the I/O occupation grew when using 10 processors for speedup (Fig. 13b). This is mainly because the increase of computation capabilities gave rise to a computational speedup, but not for I/O performance. Thus for the DI computing problem applications, the I/O consumption turns out to be a significant issue, especially when more processors are employed.

7. Conclusion

Remote Sensing applications are commonly regarded as data-intensive issues. With the further advances in the high-resolution Earth Observation, the data volume and the data transmission rate of the satellite data downlink streams would grow dramatically. In the near future, there would be increasing demands for the global environmental and resource monitoring as well as the large scientific researches on earth sciences with even more critical requirement of time. These kind of applications would exploring much larger amount of RS data covering long time span and even larger region of earth surface. When scaling to large scale and long time span, there would be the dramatically increase of data dimensionality and data complexity together with even higher computing complexity lies in the processing these data. All these issues would inevitably make the data-intensive computing problems in the Remote Sensing data processing far worse. Until now, however, there has been no promising definition or model to quantitatively describe the DI issue properly, especially for application domains. Actually, most disciplines have their own application-specific features of DI computing.

As described above, the DI_{RS} model offers a normalized DI index together with an empirical model for quantitatively estimating and analyzing DI issues occurring in RS data processing. Instead of solely depending on a single performance index, we combined several potential factors to formulate a comprehensive index. Combining empirical knowledge with experimental analysis of the algorithms throughout the whole RS data processing flow, we built an empirical model (DI_{RS}) for this DI index. The experimental results show that the DI_{RS} model is capable of describing and identifying the DI issues across various applications.

Despite of the differences in the spatial, spectral and temporal resolution across various satellites like BJ-1, SPOT4/5 and ZY02/03, the satellite data processing flows and the algorithms of each processing stage are relatively similar. So, this DI_{RS} model could also be applicable to other satellite processing systems for the quantified estimation of the DI issue. To draw the conclusion that the DI_{RS} model is a simple but promising way of estimating data-intensive issues in Remote Sensing data processing.

Acknowledgments

Dr. Lizhe Wang's work is supported by the National High Technology Research and Development Program of China ("863" Program) (No. 2013AA12A301) and the National Natural Science Foundation of China under Grant (No. 61361120098). Dr. Yan Ma's work is supported by the Preferred Foundation of Director of Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, PR China.

References

[1] Data-management challenge, Technical Report 1–2, U.S. Department of Energy, March–May 2004.

[2] Collin Bennett, Robert L. Grossman, David Locke, Jonathan Seidman, Steve Vejcik, Malstone: towards a benchmark for analytics on large data clouds, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, US, 2010, pp. 145–152.

[3] Robert Bindschadler, Patricia Vornberger, Andrew Fleming, Adrian Fox, Jerry Mullins, Douglas Binnie, Sara Jean Paulsen, Brian Granneman, David Gorodetzky, The landsat image mosaic of antarctica, Remote Sens. Environ. 112 (12) (2008) 4214–4226.

[4] Mario Cannataro, Domenico Talia, Pradip K. Srimani, Parallel data intensive computing in scientific and commercial applications, Parallel Comput. 28 (5) (2002) 673–704.

[5] F. Checconi, F. Petrini, Massive data analytics: the graph 500 on ibm blue gene/q, IBM J. Res. Dev. 57 (1) (2013) 111–121.

[6] E. Christophe, J. Michel, J. Inglada, Remote sensing processing: from multicore to gpu, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 4 (3) (2011) 643–652.

[7] E. Christophe, J. Michel, J. Inglada, Remote sensing processing: from multicore to gpu, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 4 (3) (2011) 643–652.

[8] Teresa Davies, Christer Karlsson, Hui Liu, Chong Ding, Zizhong Chen, High performance linpack benchmark: a fault tolerant implementation without checkpointing, in: Proceedings of the International Conference on Supercomputing, ICS '11, ACM, New York, NY, USA, 2011, pp. 162–171.

[9] G. De Grandi, P. Mayaux, Y. Rauste, A. Rosenqvist, M. Simard, S.S. Saatchi, The global rain forest mapping project jers-1 radar mosaic of tropical africa: development and product characterization aspects, IEEE Trans. Geosci. Remote Sens. 38 (5) (2000) 2218–2233.

[10] Meixia Deng, Liping Di, Genong Yu, A. Yagci, Chunming Peng, Bei Zhang, Dayong Shen, Building an on-demand web service system for global agricultural drought monitoring and forecasting, in: 2012 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2012, pp. 958–961.

[11] Mathieu Fauvel, Jon Atli Benediktsson, John Boardman, John Brazile, Lorenzo Bruzzone, Gustavo Camps-Valls, Jocelyn Chanussot, Paolo Gamba, A Gualtieri, M Marconcini, et al, Recent advances in techniques for hyperspectral image processing, Remote Sens. Environ. (2007) 1–45.

[12] P. Gamba, Peijun Du, C. Juergens, D. Maktav, Foreword to the special issue on human settlements: a global remote sensing challenge, IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens. 4 (1) (2011) 5–7.

[13] Phil Garcia, Multithreaded architectures and the sort benchmark, in: Proceedings of the 1st International Workshop on Data Management on New Hardware, DaMoN '05, ACM, New York, NY, USA, 2005, pp. 1–10.

[14] I. Gorton, Software architecture challenges for data intensive computing, in: Seventh Working IEEE/IFIP Conference on Software Architecture, 2008, WICSA 2008, February 2008, pp. 4–6.

[15] Huadong Guo, Jianbo Liu, An Li, Jianguo Zhang, Earth observation satellite data receiving, processing system and data sharing, Int. J. Digital Earth 5 (3) (2012) 241–250.

[16] Huadong Guo, Changlin Wang, Building up national earth observing system in china, Int. J. Digital Earth 6 (3) (2005) 167–176.

[17] Stephen D. Miller, Kerstin Kleese van Dam, Dongsheng Li, Challenges in Data Intensive Analysis at Scientific Experimental User Facilities, Springer, Berlin, 2011. pp. 249–284.

[18] R.T. Kouzes, G.A. Anderson, S.T. Elbert, I. Gorton, D.K. Gracio, The changing paradigm of data-intensive computing, Computer 42 (1) (2009) 26–34.

[19] Shunlin Liang, Xiang Zhao, Suhong Liu, Wenping Yuan, Xiao Cheng, Zhiqiang Xiao, Xiaotong Zhang, Qiang Liu, Jie Cheng, Hairong Tang, Yonghua Qu, Yancheng Bo, Ying Qu, Huazhong Ren, Kai Yu, John Townshend, A long-term global land surface satellite (glass) data-set for environmental studies, Int. J. Digital Earth 6 (2013) 5–33.

[20] Minxia Liu, Jianwen Ai, Tongkai Ji, Workflow process service research based on cloud computing platform for remote sensing quantitative retrieval, in: 2012 2nd International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), June 2012, pp. 1–4.

[21] Yan Ma, Lizhe Wang, Dingsheng Liu, Peng Liu, Jun Wang, Jie Tao, Generic parallel programming for massive remote sensing data processing, in: 2012 IEEE International Conference on Cluster Computing (CLUSTER), September 2012, pp. 420–428.

[22] Yan Ma, Lizhe Wang, A.Y. Zomaya, Dan Chen, R. Ranjan, Task-tree based large-scale mosaicking for massive remote sensed imageries with dynamic dag scheduling, IEEE Trans. Parallel Distrib. Syst. 25 (8) (2014) 2126–2137.

[23] Reagan Moore, Thomas A. Prince, Mark Ellisman, Data-intensive computing and digital libraries, Commun. ACM 41 (11) (1998) 56–62.

[24] Antonio J. Plaza, Special issue on architectures and techniques for real-time processing of remotely sensed images, J. Real-Time Image Process. 4 (3) (2009) 191–193.

[25] Antonio J. Plaza, Chein-I. Chang, High Performance Computing in Remote Sensing, Chapman & Hall/CRC, 2007. Number 1–9.

[26] Hampapuram Ramapriyan, Jennifer Brennan, Jeanne Behnke, Managing Big Data: Nasa Tackles Complex Data Challenges, Website, 2013.

[27] A. Rosenqvist, M. Shimada, B. Chapman, K. McDonald, G. De Grandi, H. Jonsson, C. Williams, Y. Rauste, M. Nilsson, D. Sango, M. Matsumoto, An overview of the jers-1 sar global boreal forest mapping (gbfm) project, in: Geoscience and Remote Sensing Symposium, 2004, IGARSS '04, Proceedings, 2004 IEEE International, vol. 2, September 2004, pp. 1033–1036.

[28] Jawwad Shamsi, Muhammad Ali Khojaye, Mohammad Ali Qasmi, Data-intensive cloud computing: requirements, expectations, challenges, and solutions, J. Grid Comput. 11 (2) (2013) 281–310.

[29] Qiao Wang, Technical system design and construction of china's hj-1 satellites, Int. J. Digital Earth 5 (3) (2013) 202–216.

[30] Le Yu, Jie Wang, Nicholas Clinton, Qinchuan Xin, Liheng Zhong, Yanlei Chen, Peng Gong, From-gc: 30 m global cropland extent derived through multisource data integration, Int. J. Digital Earth 6 (6) (2013) 521–533.

[31] Van Zyl, Jakob, Application of satellite remote sensing data to the monitoring of global resources, in: Technology Time Machine Symposium (TTM), 2012 IEEEl, 2012, pp. 1.

Please cite this article in press as: Y. Ma et al., Towards building a data-intensive index for big data computing – A case study of Remote Sensing data processing, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.10.006