FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 394-398 | e-ISSN: 2347-470X

A High-Performance Infrastructure for Remote Sensing Data Applications Using HPC Paradigms

W. T. Chembian[1], D. Hemanand[2], A. Thomas Paul Roy[3], P. Deepak Franklin[4], G. Krishna Kumari[5], N. C. Sendhil Kumar[6], S. V. Hemanth[7] and A. Gnana Soundari[8]

[1]*Associate Professor, Department of Computer Science and Engineering, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College (Autonomous), Avadi – Vel Tech Road Vel Nagar Avadi, Chennai, Tamil Nadu, India*
[2]*Professor, Department of Computer Science and Engineering, S.A. Engineering College (Autonomous), Thiruverkadu, Chennai-600077, Tamil Nadu, India*
[3]*Professor, Department of Computer Science and Engineering, PSNA College of Engineering and Technology, Kothandaraman Nagar, Tamil Nadu, India*
[4]*Assistant Professor, Department of Electrical and Electronics Engineering, Sriram Engineering College, Perumalpattu, Veppampattu (R.S.) Tiruvallur,Tamil Nadu, India*
[5]*Professor, Department of Humanities and Science (Mathematics),Vardhaman College of Engineering, Ranga Reddy, Telangana*
[6]*Professor, Department of Electronics and Communication Engineering, Sri Indu College of Engineering & Technology, Sheriguda, Hyderabad, Telangana, India*
[7]*Assistant Professor, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Medchal-Malkajgiri, Hyderabad, Telangana, India*
[8]*Professor, Department of Information Technology, Jeppiaar Engineering College, Chennai- 600119, Tamil Nadu, India*

*__Corresponding Author:__ D. Hemanand; E-mail: d.hemanand@gmail.com

ABSTRACT- Individuals and businesses are currently involved in the administration of remote sensing data that was previously handled only by government agencies. There is a lot more information in remote sensing data than go through the eye, and retrieving it is time-consuming and computationally expensive. Clusters, distributed networks, and specialized hardware devices are essential to speeding up remote sensing data extraction calculations. HPC advances in remote sensing applications are examined in this research. High-performance computing (HPC) concepts for instance FPGAs and GPUs as well as large-scale and heterogeneous computer networks are examined (GPUs). Using HPC paradigms, remote sensing applications are examined in these sections.

Keywords: Remote sensing, Distributed Computing, High Performance Computing, Accelerators.

ARTICLE INFORMATION
Author(s): W. T. Chembian, D. Hemanand, A. Thomas Paul Roy, P. Deepakfranklin, N. C. Sendhil Kumar, G. Krishna Kumari, S. V. Hemanth and A. Gnana Soundari

Special Issue Editor: Dr. S. Gopalakrishnan ;
Received: 10/05/2022; **Accepted**: 14/06/2022; **Published**: 30/06/2022;
E- ISSN: 2347-470X;
Paper Id: 0322SI-IJEER-2022-24;
Citation: 10.37391/ijeer.100255
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100255.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

This article belongs to the Special Issue on **Intervention of Electrical, Electronics & Communication Engineering in Sustainable Development**

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Sensing and computer technologies are transforming remote sensing data gathering, administration, and analysis. Latest-generation sensor technology in Earth and space observation systems has led to a steady supply of high-dimensional data, causing new processing challenges. Et al (2009). Carvajal et al. rely on computationally effective approaches to turn remote sensing data into scientific knowledge (2005).

As the number of firms and people employing remote sensing data rises, so does the need for efficient distribution methods. Chang/Hsueh (2008). Recent research has focused on incorporating HPC tools and methods into remote sensing missions. As an integrated collection of computing environments and programming methodologies, HPC may help with large-scale difficulties like those in remote sensing [4].

Earth, space, and exploration science remote sensing requests claim real-time or near-real-time data interpretation. Environmental research, military uses, and monitoring threats including wildfires, oil spills, and chemical/biological contamination are examples [5]. HPC systems are being used in remote sensing. Computer scientists' usage of COTS computer equipment in "teams" has fostered new breakthroughs based on multi-processor machines [6]. *Figure 1* depicts a remote sensing system.

HPC system design for data-intensive issues has been shifting from homogeneous processing units to heterogeneous computing resources during the last decade [7]. It is the outcome of technological advancement and shifts in the computer industry [8] that give rise to such variation. Remote sensing applications may benefit greatly from heterogeneous

Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 394-398 | e-ISSN: 2347-470X

COTS resources, therefore grid and cloud computing are attempting to make these platforms easier to utilise for heterogeneous and distributed computing.

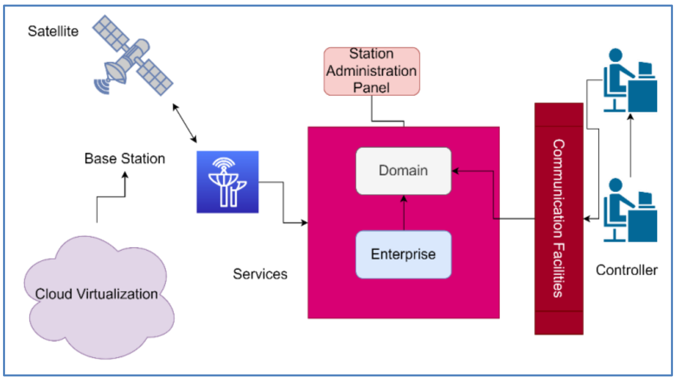

Figure 1. Overview of a remote sensing system

Using these technologies, it is simpler to transfer and handle data with a high dimension. Cluster centres or infrastructures of CPUs are ideal for remote sensing data processing, but for in-situ remote sensing data processing, where low-weight and low power integrated components are essential to reduce mission message and obtain assessment results in real time, i.e. at the same time information is collected, these systems are prohibitively expensive and difficult to implement.

FPGAs and GPUs connect on-board and real-time distant sensing data interpretation. Remote sensing applications may use the compact size and low cost of these hardware components to boost processing. Large payloads are needed for remote sensing [10].

2. LITERATURE SURVEY

Airborne and satellite sensor systems are increasingly being equipped with remote sensing-related technologies [11]. Preprocessing data and delivering only data that fulfils content requirements when on-board processing is enabled may reduce downlink bandwidth consumption at the sensor. On-board processing may better use expensive hardware. The capacity to acquire, analyse, and act on images quickly is another advantage [12].

On-board processing may reduce data transfer rate. Onboard processing makes ground processing systems more accessible [13]. These processing advances will assist a broad variety of remote sensing applications, including Earth observation missions that are exploring adding specialist hardware components, web sensors, and planetary exploration missions that can make autonomous choices on board [14].

There has been a lot of focus on GPUs and FPGAs in recent years. Recent studies contrasted the two methods of processing hyperspectral images acquired by remote sensing. Remote sensing helped to develop cluster computing, a valuable parallel computing solution, to meet computational requirements. Desktop PCs couldn't handle the processing power of Earth observation sensor devices. Previous CPU advancements aimed on improving the number of clock cycles per core [15].

Most computers today have energy-hungry multi-core CPUs. Remote sensing uses these platforms [16]. Many remote sensing applications have significant computational needs, which multi-core CPU parallel systems may not be able to meet. Clusters with hardware accelerators are a recent commodity computing discovery.

GPUs were formerly utilised just for graphics until recently [17]. Thanks to advances in GPU hardware and software, especially NVidia CUDA, GPUs are gaining popularity in scientific computing. GPUs improve cluster-based systems' processing power [18]. FPGAs and other accelerators have developed from pricey application-specific equipment to highly parallel, programmable commodity components [19].

NVidia's GPUs can deliver up to 525 Gigaflops due to twice peak efficiency, which is quicker than the fastest four core CPU [20]. Remote sensing applications may leverage GPU clusters and more cores per cluster node to fulfil their growing computational needs. Specialized modifications and/or multiple core nodes are common in new cluster systems [21]. Remotely sensed data processing, storage, and administration may improve soon.

3. PROPOSED SYSTEM

Virtualization is a major benefit. Middleware functions as a virtual organisation between hardware design and user interface idea. The goal is to alleviate users of the stress of maintaining workflow resources so they may focus on scientific activity. Users may theoretically identify, access, and combine data and computational resources as part of their workflow.

This capacity must be on-demand and meet performance and timeliness criteria. Virtualization may improve computational resources and data sources as sensors and instruments. Virtualized sensors may be used for remote and on-site measurements. A user may provide a spatial-temporal bounding box with geographic and frequency resolution and sampling criteria, and the system would then pick the optimal detector using "natural" terminology and semantics. Figure 2 depicts the system architecture.

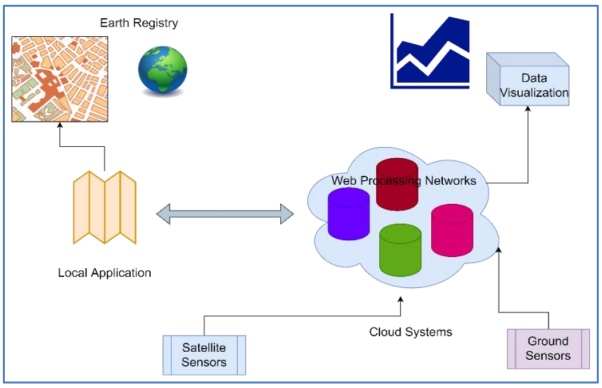

Figure 2. Proposed System Architecture

Virtualization also lets users search for measurements and observations based on metadata. You may search using geographical and temporal bounding boxes, sample characteristics, and a quantitative or geophysical parameter. Virtualization allows upgrades without affecting availability. DCI promotes interoperability by using standard protocols and interfaces, such as SOA.

These standards offer composability and extensibility, which are essential to meet performance objectives. DCI is reusable and easy to implement. As stated later in this section, many remote sensing data applications are event-driven and don't need continual system resources. In an emergency, like a hurricane or earthquake, it's crucial to have supplies on hand. Until full capacity is needed, this system uses little resources to stay ready. Because these capabilities are not idle, they may be utilised by other programmes and only accessed when required, using a priority-based use approach that priorities crises above scientific pursuits. Using service architectural concepts, frameworks, and technologies, you may create Problem-Solving Environments (PSE).

PSEs give the basis for tackling specific scientific difficulties. The framework delivers tools in the scientific area's language so users may handle them with minimum learning. This framework encapsulates powerful data-processing capabilities. Managing component APIs supports many message bus implementations.

Tibco SmartSockets, Tibco Roadhouse, Interface Control System's Program Bus, and the Elvin distributed event routing service may be used. GMSEC may employ commercial messaging features including publish-subscribe, assured delivery, and security. On-orbit systems like those in GMSEC Reference Architecture may utilise message buses. Firewalls or, on versions, the actual ground link separate multiple message bus instances. Portal server components limit data access to external web servers. Portal server and firewall regulate any traffic returning to the communication bus network.

4. RESULTS AND DISCUSSION

Experiments used four parallel computing platforms: Thunderhead is a NASA cluster with 256 nodes, 1GB RAM, and 80GB main memory.

Table 1. Execution Cycle time

Processor Number	Cycle Time
P1	0.2547
P2	0.32547
P3	0.12547
P4	0.02354
P5	0.02147
P6	0.0357
P7	0.06547
P8	0.02547
P9	0.09854
P10	0.2478

NASA's Goddard Space Flight Center in Maryland connects it via 2 GHz optical cable Myrinet. This network has 16 workstations and four methods to reach them. Before assessing parallel performance, we evaluate endmember extraction and spectral un-mixing accuracy in the context of the relevant application. *Table 1* lists cycle time results.

To analyse and evaluate results, define the un-mixing chain's parameter values. After determining the dataset size using virtual dimensionality, all of our implementations configured to eliminate endmembers. Skewer count was chosen at because, while numbers of and were also tested, using resulted to the loss of essential end members, whereas end members obtained using were virtually similar to those found using. *Figure 3* analyses processor-based execution time.

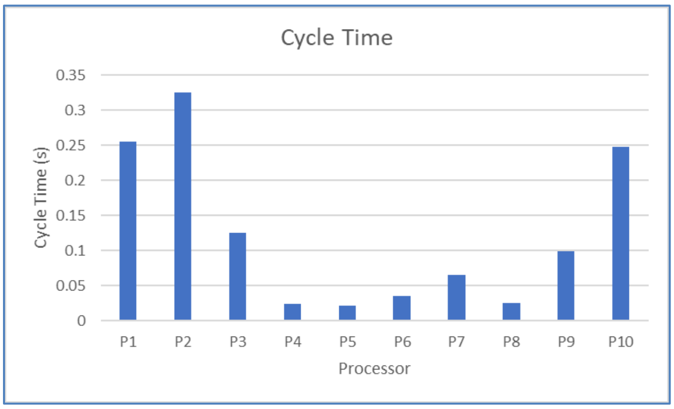

Figure 3. Processor based execution time analysis

After entering all parameters, the threshold angle and cut-off threshold value were set to the iteration mean as a limit of tolerance. These parameter settings are comparable to those in the literature. This assignment evaluates four simultaneous implementations to estimate the sub-pixel frequency of fires in the WTC scene using USGS heat spot baseline information. *Table 2* displays hyperspectral analysis findings.

Table 2. Hyperspectral analysis

Identified Spots	Cluster Area	FPGA Area
S1	0.51	0.147
S2	0.025	0.2458
S3	0.34	0.1478
S4	0.54	0.2549
S5	0.66	0.325
S6	0.74	0.425
S7	0.0124	0.654
S8	0.578	0.147

Since each pixel of AVIRIS data is 1.7 square meters, thermal hot spots are sub-pixel in nature. Parallel approaches may reliably predict thermal hotspots, leading to comparable results as ENVI's wavelet transform chain. Figure 4 shows FPGA and cluster hyperspectral analysis.

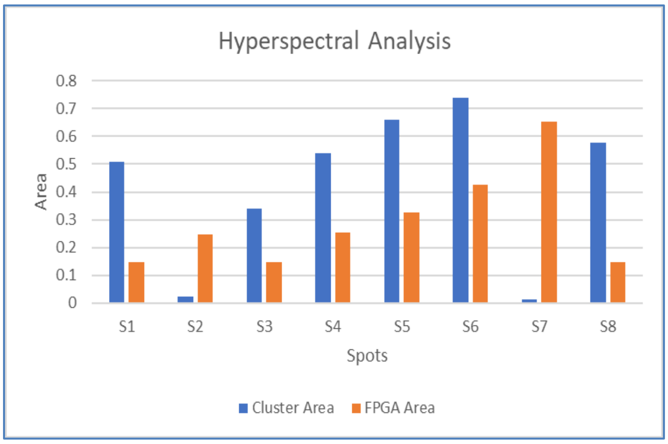

Figure 4. Hyperspectral analysis for FPGA and cluster area

Thermal hotspot estimates were accurate and consistent across all four deployments. Parallel approaches' output was validated using both our own serial implementations and ITTV's commercial version 4.5 of ENVI software, which employs the same parameters.

Parallel implementations found identical endmembers to serial adaptations in both circumstances. We used the same random skewers to ensure serial communication versions were identical. Parallel implementations gave different endmembers than ENVI. *Table 3* lists cluster-based outcomes.

Table 3. Cluster based Outcomes

Number of CPUs	Cluster Model	Heterogeneous model
1	285.254	75.86
4	214.036	68.65
16	189.025	39.25
36	299.32	54.47
64	321.054	68.37
100	333.021	78.32
144	378.254	77.36
196	199.524	80.2
256	155.0214	84.65

Our investigations demonstrate that the basic aspect values between parallel and original endmembers are frequently below 1.75, indicating that the resultant endmembers sets are spectrally comparable. Also shown is how long a single Thunderhead node needs to perform an optimised serial processing chain. Both sequential and parallel computation speeds for the root node's computations were evaluated. There are also downtimes. *Figure 5* depicts cluster analysis result model.

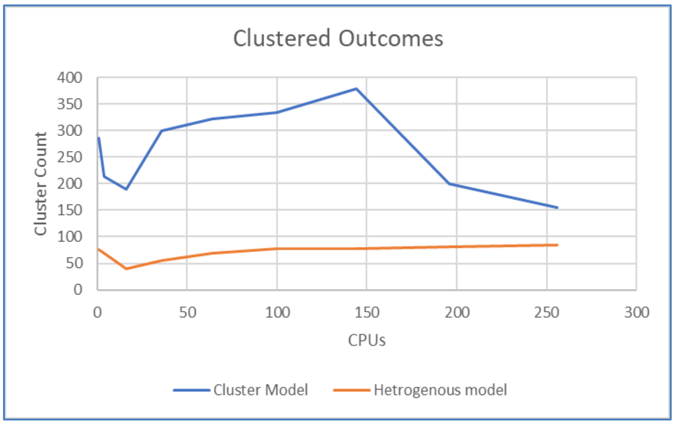

Figure 5. Cluster analysis outcome model

Timings and how often parallel was faster than sequential. Every Thunderhead processor took 2271.22 seconds to perform the serial code, however the fastest heterogeneous network processor took 1894.02 seconds. Figure 6 shows CPU memory analysis.

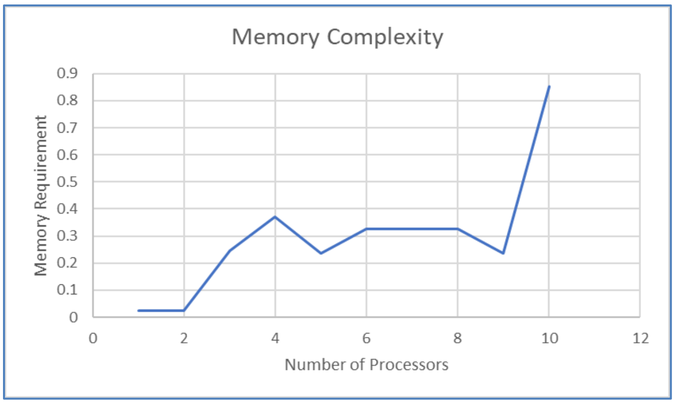

Figure 6. Memory requirement analysis

Comparing the timeframes for sequential and parallel calculations on cluster and heterogeneous networks showed great parallel efficiency. *Figure 7* depicts processor communication capabilities.

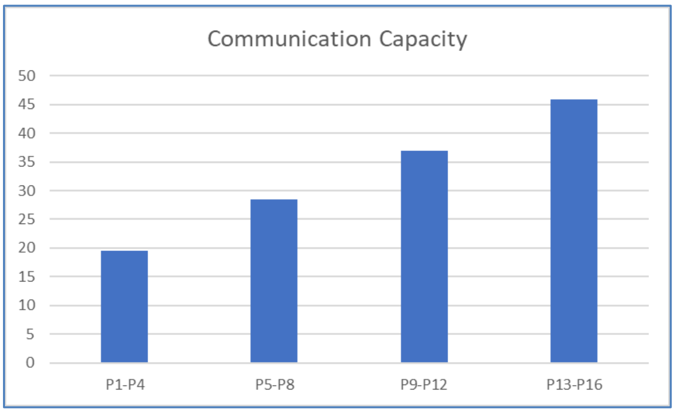

Figure 7. Processor range-based communication capacity

5. CONCLUSION

With regard to hyperspectral remote sensing issues, the performance of several parallel computing architectures for hyperspectral un-mixing chains has been evaluated in this study. Parallel solutions for heterogeneous networks, commodity clusters, FPGA-based solutions, and GPU-based approaches are all discussed. According to our findings, remote sensing picture sets that have never been analysed after being collected may benefit from the processing capacity provided by heterogeneous networks and clusters. These data may provide useful parameters in other disciplines. To meet the timing constraints of many hyperspectral remote sensing applications, we have additionally created FPGA and GPU-based prototypes of the proposed hyperspectral un-mixing chain.

ACKNOWLEDGMENT

The author would like to appreciate the effort of the editors and reviewers. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] Benediktsson, A. Plaza, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, J. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, “Recent advances in techniques for hyperspectral image processing,” Remote Sens. Environ., vol. 113, pp. 110–122, 2009.
[2] Carvajal, W. Rivera, and W. Lugo, “Service oriented architecture grid-based environment for hyperspectral imaging analysis,” Int. J. Inf. Technol., vol. 11, no. 4, pp. 104–111, 2005.
[3] Chang and Hsueh, “Field programmable gate arrays (FPGA) for pixel purity index using blocks of skewers for endmember extraction in hyperspectral imagery,” Int. J. High Performance Comput. Applicat., vol. 22, pp. 408–423, 2008.
[4] Chang and A. Plaza, High Performance Computing in Remote Sensing. Boca Raton, FL: CRC Press, 2007.
[5] Dongarra and A. Lastovetsky, High-Performance Heterogeneous Computing. New York: Wiley, 2009.
[6] Fisk, R. Brightwell, D. Greenberg, T. Hudson, M. Levenhagen, A. Maccabe, and R. Riesen, “Massively parallel computing using commodity components,” Parallel Comput., vol. 26, p. 243266, 2000.
[7] Goldberg, L. Zhou, C. Barnet, Z. Cheng, F. Chung, W. Wolf, T. King, X. Liu, H. Sun, and M. Divakarla, “Regression of surface spectral emissivity from hyperspectral instruments,” IEEE Trans. Geosci. Remote Sens., vol. 46, pp. 328–333, 2008.
[8] Hauck, “The roles of FPGAs in reprogrammable systems,” Proc. IEEE, vol. 86, pp. 615–638, 1998.
[9] Itoh, “Massively parallel fourier-transform spectral imaging and hyperspectral image processing,” Opt. Laser Technol., vol. 25, p. 202, 1993.
[10] Kaewpijit, T. El-Ghazawi, and J. L. Moigne, “Parallel and adaptive reduction of hyperspectral data to intrinsic dimensionality,” Cluster Comput., vol. 1, pp. 102–110, 2001.
[11] Lastovetsky, D. Valencia, M. O’Flynn, A. Plaza, and J. Plaza, “Parallel processing of remotely sensed hyperspectral images on heterogeneous networks of workstations using HeteroMPI,” Int. J. High Perform. Comput. Applicat., vol. 22, no. 4, pp. 386–407, 2008.
[12] Nekovei and Du “Fast real-time onboard processing of hyperspectral imagery for detection and classification,” J. Real-Time Image Process., vol. 22, pp. 438–448, 2009.
[13] Neville, Brazile, K. Staenz, D. Schlaepfer, L. Sun, and K. I. Itten, “Cluster versus grid for operation generation of ATCOR’s MODTRAN-based look up table,” Parallel Comput., vol. 34, pp. 32–46, 2008.
[14] Nickolls, E. Lindholm, S. Oberman, and J. Montrym, “NVIDIA Tesla: A unified graphics and computing architecture,” IEEE Micro, vol. 28, pp. 39–55, 2008.
[15] Palencia, J. Dorband, and U. Ranawake, “Commodity computing clusters at goddard space flight center,” J. Space Commun., vol. 3, p. 1, 2003.
[16] Plaza, A. Plaza, and D. Valencia, “Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data,” J. Supercomput., vol. 40, no. 1, pp. 81–107, 2007.
[17] Rosenbaum, U. Thomas, F. Kurz, S. Suri, and P. Reinartz, “A new software/hardware architecture for real time image processing of wide area airborne camera images,” J. Real-Time Image Process., vol. 5, pp. 229–244, 2009.
[18] Valencia, Plaza, J. Plaza, and P. Martinez, “Commodity clusterbased parallel processing of hyperspectral imagery,” J. Parallel Distrib. Comput. vol. 66, no. 3, pp. 345–358, 2006.
[19] Valencia, Plaza, and J. Plaza, “An experimental comparison of parallel algorithms for hyperspectral analysis using homogeneous and heterogeneous networks of workstations,” Parallel Comput., vol. 34, no. 2, pp. 92–114, 2008.
[20] Zhang, S. Kalluri, J. JaJa, S. Liang, and J. Townshend, “Characterizing land surface anisotropy from AVHRR data at a global scale using high performance computing,” Int. J. Remote Sens., vol. 22, pp. 2171–2191, 2001.
[21] Zhao, S. Tehranian, T. Harvey, A. Swaroop, and K. Mckenzie, “A robust framework for real-time distributed processing of satellite data,” J. Parallel Distrib. Comput., vol. 66, pp. 403–418, 2006.