

## Research Article

# Qualitative Precipitation Estimation from Satellite Data Based on Distributed Domain-Specific Architecture

Sethakarn Prongnuch <sup>1</sup>, Theerayod Wiangtong,<sup>2</sup> and Suchada Sitjongsataporn<sup>3</sup>

<sup>1</sup>Department of Computer Engineering, Faculty of Industrial Technology, Suan Sunandha Rajabhat University, Bangkok 10300, Thailand

<sup>2</sup>Department of Electrical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand

<sup>3</sup>Department of Electronic Engineering, Mahanakorn Institute of Innovation (MII), Faculty of Engineering and Technology, Mahanakorn University of Technology, Bangkok 10530, Thailand

Correspondence should be addressed to Sethakarn Prongnuch; [sethakarn.pr@ssru.ac.th](mailto:sethakarn.pr@ssru.ac.th)

Received 18 September 2020; Revised 16 January 2021; Accepted 1 February 2021; Published 16 February 2021

Academic Editor: Angelos Markopoulos

Copyright © 2021 Sethakarn Prongnuch et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the qualitative precipitation estimation (QPE) based on data from the Himawari satellite and distributed-domain specific architecture. The QPE process consists of receiving and managing the raw data from the satellite every 10 minutes and calculating the rain-temperature relationship. The aim of this research is to decrease the QPE processing time by using distributed domain-specific architecture (DDSA), with 9 small computing boards are connected to a gigabit switch. Instead of using a high-performance PC, this distributed embedded system is also suitable for processing interval data receiving from the satellite every 10 minutes. The experimental results show that the proposed fast-satellite data processing algorithm is optimal for QPE processing on the DDSA platform, requiring 115.53 seconds processing time and low power consumption.

## 1. Introduction

The impacts from tropical cyclones include with heavy rainfall and strong wind that can cause significantly massive loss of life and infrastructure. Many efforts are being made to understand about the meteorological parameters in physical terms. To this end, the tropical cyclones' characteristics determined from satellite data are of interest for researchers.

Qualitative analysis via quantitative precipitation estimation (QPE) is a method used to estimate the amount of precipitation from specific satellite data such as the rainfall across a region. The maps used for precipitation estimation over a certain area and time span are compiled using the different data types collected by satellites. There have been many studies on QPE using satellite-based parameters. Regional rainfall event (RRE), defined in [1], was presented with the concept of QPE from satellite products compared with rain gauge observations from a stationary network of hydrometeorologi-

cal stations. The results showed that the estimates from satellite data are more reliably modelled variation of rainfall in terms of QPE within the RRE perspective. Similarly, Panda et al. [2] investigated how to determine the characteristic features of the tropical PHET cyclone in 2010 using the satellite-based meteorological parameters, including QPE, temperature on the sea surface, and relative humidity in the upper troposphere, compared with numerical model simulations using a model of the Weather Research and Forecasting (WRF) system. These satellite-derived parameters have been shown to be suitable for determining the meteorological conditions of tropical cyclones. Adler et al. [3] confirmed that satellite-based precipitation observations over the last 30 years have presented more accurate estimates of rainfall for understanding regional and global climates. As stated in [4], the precipitation estimates from satellite archives for decades over the sea surface are significant in learning the characteristics of ocean-atmosphere interaction.

TABLE 1: Research work on remote sensing application.

Study area	Approach	Technique
Anchor-free single stage detector [5]	Multiscale dense path aggregation feature pyramid network	High-level semantic information and low-level location information
Multiscale object detection [6]	Region-based convolutional neural networks (RCNNs) and feature pyramid network	Adaptive anchor networks
Automatic aggregation [7]	Hierarchical similarity diffusion	To measure the similarity matrix of image in the data set cluster by cluster
Monitoring forest dynamics [8]	Spatial coordinates correction approach	Registering low-altitude UAV visible light and hyperspectral images

Regarding remote sensing data, the object detector relied on the anchor boxes for remote sensing imagery introduces a large number of hyperparameters and increases the computational redundancy of the detection model. Avoiding the large number of hyperparameters, an anchor-free single stage detector has been proposed based on multiscale dense path aggregation [5]. An idea of adaptive anchor networks for multiscale object detection has been presented with a feature pyramid [6]. Based on the intelligent annotation approach, an automatic aggregation via hierarchical similarity diffusion has been modified [7]. A dynamic monitoring technology using a spatial coordinates correction approach [8] has been presented by using low altitude unmanned aerial vehicle (UAV) visible light and hyperspectral images for monitoring forest dynamics. So, the research work on the remote sensing applications can be summarized in Table 1.

Earth observation imagery taken by imaging satellites is used for satellite data processing to supply environmental information about the surface temperature and weather changes around the world. An advantage of distributed domain-specific architecture (DDSA) is its ability to solve specific problems such as tasks under limited resources using FPGA-based DDSA architecture [9], high-performance computing tasks using GPUs [10], and low-power tasks using embedded systems [11].

In Thailand, the Thailand Meteorological Department (TMD) [12] is the government organization which provides the weather forecasting and warnings to mitigate natural disasters. DDSA [9] switches processing from uniprocessors to multiprocessors. While computers with standard processors run conventional programs through the operating system, DDSA can perform many small tasks extremely well. It could be much more heterogeneous than recent homogeneous multicore chips [10]. In Thailand, the TMD and Ministry of Digital Economy and Society have collaborated with Hewlett Packard Enterprise (HPE) to provide high-performance computing (HPC) solutions such as the HPE APOLLO 2000 GEN10 system [13] in order to enhance the performance for forecasting the weather and predicting the natural disasters with the high accuracy [14].

In this paper, we present the deployment and evaluation of distributed domain-specific architecture for qualitative precipitation estimation using satellite data. Our main objective is to present rainfall estimation using visualizations of tropical cyclone movement and rain over Thailand for public warning. It is convenient to explain weather phenomena in

the form of cyclone motion visualizations for those who are not meteorologists.

## 2. Background and Related Works

The works related to this paper can be separated into two parts in terms of qualitative precipitation estimation from satellite data and embedded heterogeneous system (EHS) architectures on heterogeneous systems. Precipitation is a meteorological phenomenon that affects human lives and can cause economic loss. Quantitative predictions of precipitation have applied rain gauges, satellite data, and numerical models such as the advanced meteorological imagers onboard geostationary satellites for quantitative precipitation nowcasts [15]. Surveys of global precipitation variations using satellite and surface gauge information have been analyzed and reviewed by the Global Precipitation Climatology Project [13]. Improvements in global precipitation estimation have been observed between high-frequency brightness temperature depression and surface precipitation using satellite passive microwave retrievals, as detailed in [16].

The real-time parallel applications [17] on heterogeneous distributed embedded systems in an energy-efficient scheduling algorithm were composed of CPUs, memory (RAM) and nonvolatile memory, and a network interface card, which all are connected by a controller area network (CAN) bus. A task executed in one processor sends messages to all its successor tasks, which may be located on different processors implemented on a cluster consisting of six homogeneous dual-core processors (ARM Cortex-A20), 1 GB memory, and the Debian 4.7 operating system. The exploitation of approximate feature extraction via genetic programming for hardware acceleration in a heterogeneous microprocessor was presented in [18]. The architecture included an MSP430 CPU for top-level software control, a direct memory access (DMA) module for automated data movement in applications, a four-core feature extraction accelerator (FEA) for GP-model computation, a support vector machine accelerator (SVMA) for configurable SVMA classification, and accelerators and other modules memory mapped and interfaced to the CPU via a peripheral bus. An evaluating single-board computer cluster for cyber operations was proposed in [19] to study a 128-core Beowulf cluster comparing the Parallella computer architecture and Raspberry Pi architecture. A scalable object detection framework based on an

TABLE 2: Research work on heterogeneous distributed embedded systems in many areas.

Study area	Architecture	CPU/GPU	Memory	OS	Parallel technique
Energy-efficient scheduling algorithms [17]	Cluster	Cortex-A20	RAM 1GB	Debian 4.7.2-5	Directed acyclic graph (DAG)
Feature extraction via genetic programming [18]	Heterogeneous microprocessor	CPU, feature extraction accelerator (FEA), SVM	96 KB of on-chip SRAM	N/A	Vector processing
Cyber operations [19]	Cluster	ARM and epiphany	8 GB microSD card and 500 GB USB hard drive	Linux	OpenMP and MPI
Object detection framework [20]	Cluster	Dual-core ARM	32 KB	N/A	MPI
GFLink for big data [21]	Cluster	Intel Core i5-4590 CPU, NVIDIA GeForce GTX 750, NVIDIA Tesla C2050, NVIDIA Tesla 72 K20, NVIDIA Tesla P100	RAM 16 GB	N/A	Task manager

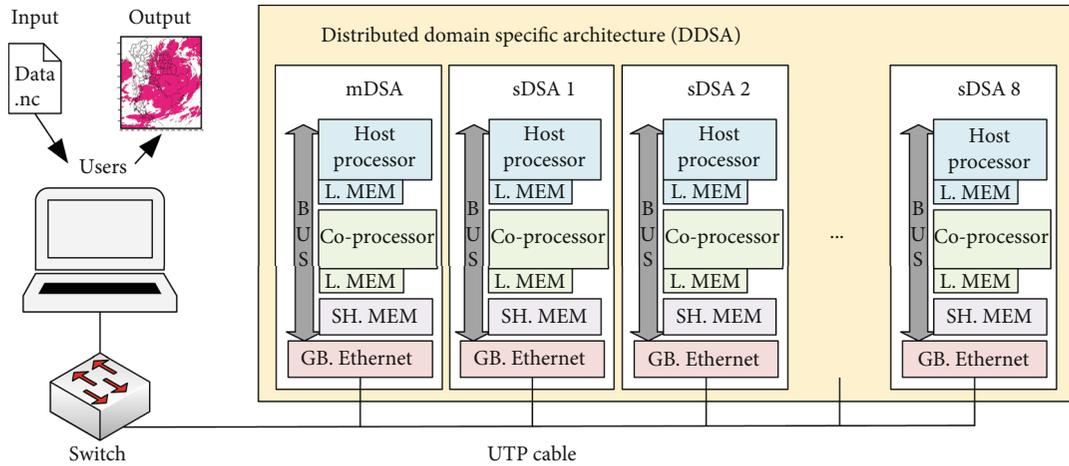


FIGURE 1: Distributed domain-specific architecture (DDSA).

embedded manycore cluster [20] was presented implemented on a Parallella board. For big-data-related applications, an in-memory computing architecture for heterogeneous CPU-GPU clusters was presented in [21], called GFLink. A performance evaluation of a system consisting of a single machine and cluster compared with an Intel Core i5-4590 CPU containing four cores running at 3.30 GHz with 16 GB memory was conducted. Several kinds of GPUs were utilized, including the NVIDIA GeForce GTX 750, NVIDIA Tesla C2050, NVIDIA Tesla K20, and NVIDIA Tesla P100. All mentioned research works on heterogeneous distributed embedded systems in many areas are summarized in Table 2.

For our research, we aimed to minimize the processing time of the QPE process using parallelism on the DDSA platform. The contributions of this study are summarized as follows:

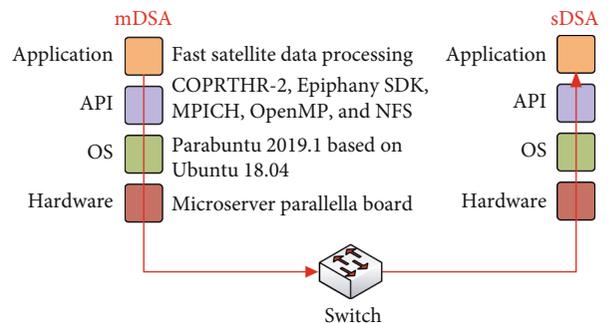


FIGURE 2: Communication from the master DSA (mDSA) to the slave DSA (sDSA).

- (i) We propose the fast satellite data processing (Fast-SDP) algorithm to minimize the overall processing time in comparisons to the previous research

```

mDSA
Host processor
1:INPUT the 9-satellite data .nc files
sDSA and mDSA
Host processor
2:CONVERT .nc file to .cdl file
3:OMP PARALLEL
Coprocessor
4:SEARCH  $temp_k$  in .cdl file
5:If  $N_{temp_c} = zero$  do  $R \leftarrow N_{temp_c}$ 
6:else  $R \leftarrow \phi \cdot 10^{\psi \cdot temp_k^{\xi}}$ 
7:end if
Host processor
8:CONVERT .cdl file to .nc file
mDSA
Host processor
9:OUTPUT the 9-satellite data .nc files
10:END

```

ALGORITHM 1: The fast satellite data processing (Fast-SDP) algorithm.

- (ii) We propose the master-slave DDSA architecture, including one master and multiple slaves of heterogeneous embedded system boards, called Parallella, connected to a gigabit switch by using the OpenMP API for parallelism

The rest of this paper is organized as follows. Section 3 presents the proposed distributed domain-specific architecture. Section 4 presents the fast satellite data processing algorithm. Section 5 details the experimental results, and Section 6 concludes this research.

### 3. Proposed Distributed Domain-Specific Architecture

Domain-specific architecture (DSA) is used to form general-purpose cores in computer architecture to improve the level of efficiency. Domain-specific algorithms are almost always for small, computationally intensive kernels of larger systems, which should focus on a specific program.

*3.1. Distributed Domain-Specific Architecture.* The proposed DDSA design is based on [9] for multiple merits: parallelism, flexibility, and scalability. Each DSA consists of a host processor with local memory (L.MEM), a coprocessor with L.MEM, a shared memory (SH.MEM), and Gigabit Ethernet interfacing (GB.Ethernet). As similar to [11], the DDSA comprises a master DSA (mDSA) and multiple slave DSAs (sDSA), connected through a star network switch, depicted in Figure 1.

*3.2. DDSA System Design.* The proposed system design consists of four layers in the standard OSI (Open Systems Interconnection) model, as shown in Figure 2; these are the hardware layer, operating system (OS) layer, application programming interface (API) layer, and application layer.

*3.2.1. Hardware Layer.* We applied the Microserver Parallella board [22] in the form of DSA. Each contains a ZYNQ SoC processor and a 16-core Epiphany RISC coprocessor. A 1 GB DDR3 memory, 32 GB microSD storage, and Gigabit Ethernet are also included.

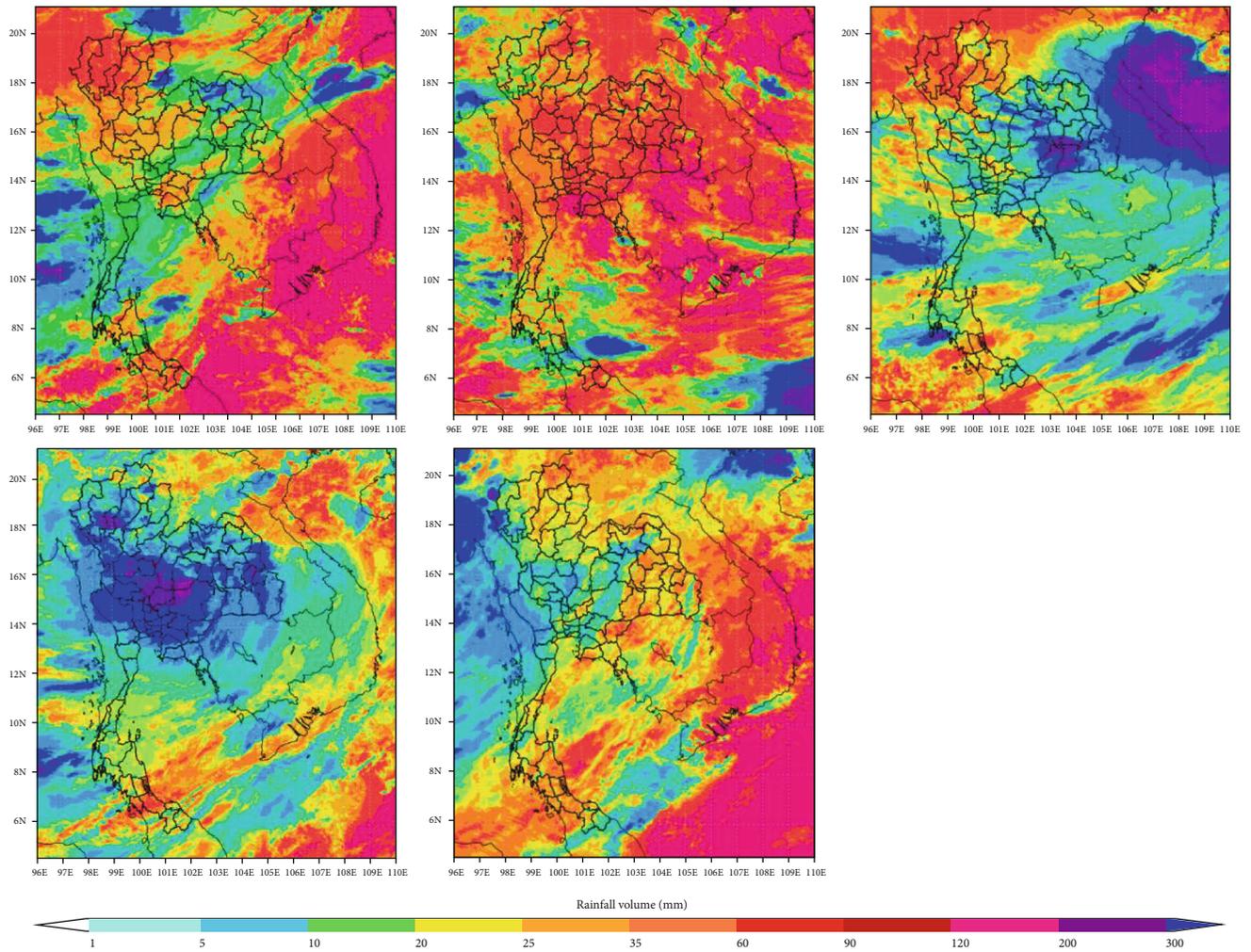
*3.2.2. OS Layer.* Parabuntu 2019.1 [23] was used as the operating system. The components of Parabuntu include a Linux 4.14 kernel based on Ubuntu 18.04, which was installed on a microSD card in the Parallella board.

*3.2.3. API Layer.* Four APIs were used for the DDSA: COPRTHR-2, Epiphany SDK, MPICH, and the Network File System (NFS) for DDSA architecture. COPRTHR-2 (The CO-PRocessing THReads) SDK version 2 [24] provides the libraries and tools, including the syscore, coprcc, coprcc-info, coprcc-db, libcoprthr-mpi, and libcoprthr. While Epiphany SDK [22] provides the library as Epiphany-C/C++ in the host processor and a coprocessor for implementation. MPICH is a widely used and portable implementation of the MPI (Message Passing Interface) standard [25] for DDSA with dynamic process management, parallel input/output, one-sided operations, and other extensions. The OpenMP API is a scalable model with a simple and flexible interface for developing parallel applications [26]. The Network File System (NFS) API utilities incorporate with the user-space server and client tools for the kernel's NFS abilities [27]. NFS is a protocol that allows for sharing file systems over the DDSA.

*3.2.4. Application Layer.* The performance of the proposed DDSA with the proposed fast satellite data processing algorithm is evaluated in the next section.

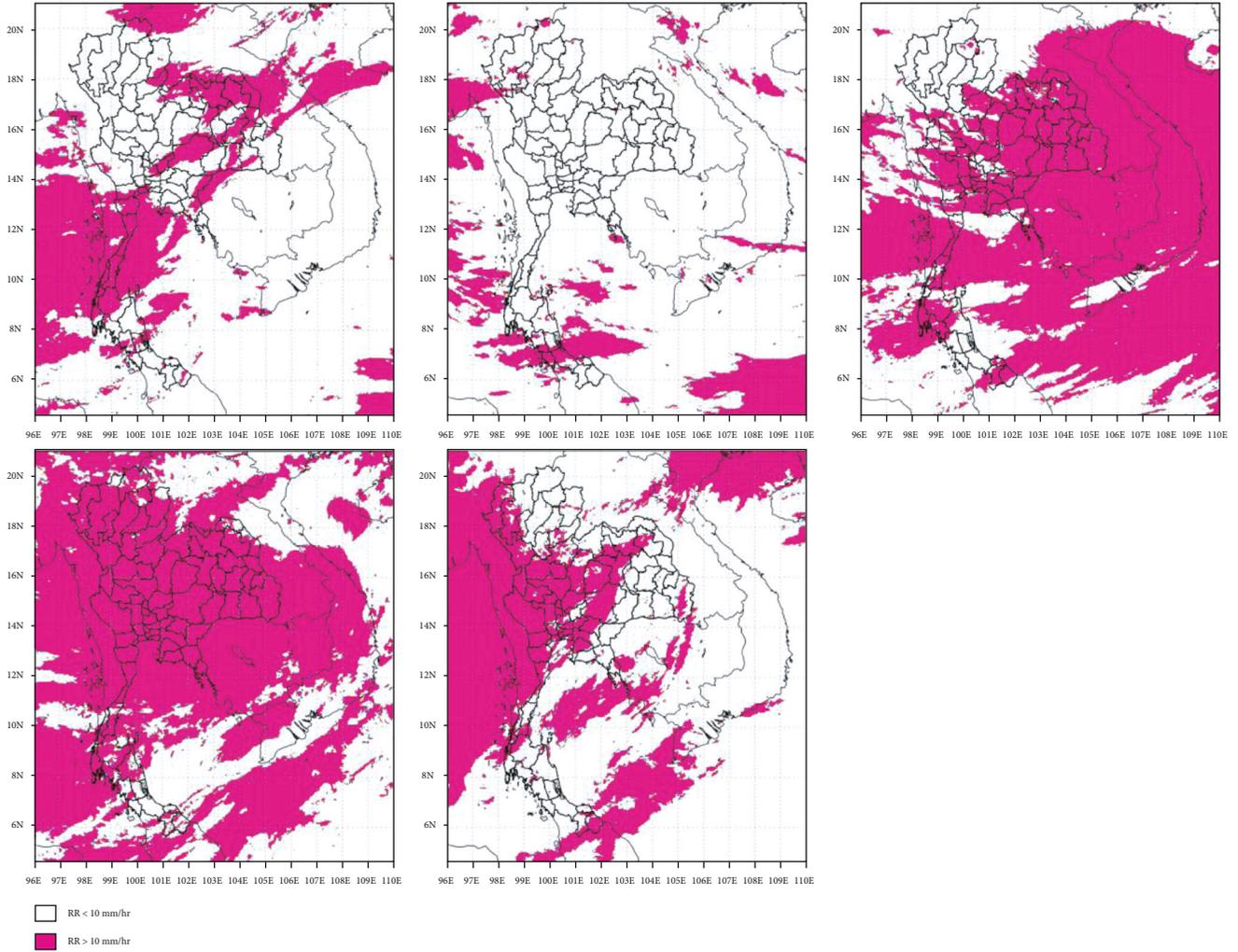
## 4. The Fast Satellite Data Processing Algorithm

In this section, we introduce the fast satellite data processing (Fast-SDP) using satellite data from the Thailand



(a) Satellite-based estimated rainfall of tropical storm *Podul* before qualitative precipitation estimation (QPE)

FIGURE 3: Continued.

(b) Satellite-based estimated rainfall of tropical storm *Podul* after QPEFIGURE 3: Satellite imageries of tropical storm *Podul* over 27-31 August 2019 processed by the proposed Fast-SDP algorithm.

Meteorological Department (TMD) [12]. Quantitative precipitation estimations of the rainfall over Thailand by the Himawari-8 satellite [28] are observed every 10 minutes. Rainfall volume is the quantity of rainfall in terms of volume measured at the base station, while the rainfall rate is obtained from translating the temperature data from the satellite. It can be used to estimate the amount of precipitation fallen with the specific data from satellites across the region.

The rainfall rate (IRR) is as a function of rainfall in terms of volume and temperature in units of mm/hr as follows [29]:

$$\text{IRR} = \phi \cdot 10^{\psi \cdot \text{temp}_k^\zeta}, \quad (1)$$

where  $\phi$ ,  $\psi$ , and  $\zeta$  denote coefficients that depended on the climates and locations. The temperature from satellite data,  $\text{temp}_k$ , was used in degrees Kelvin.

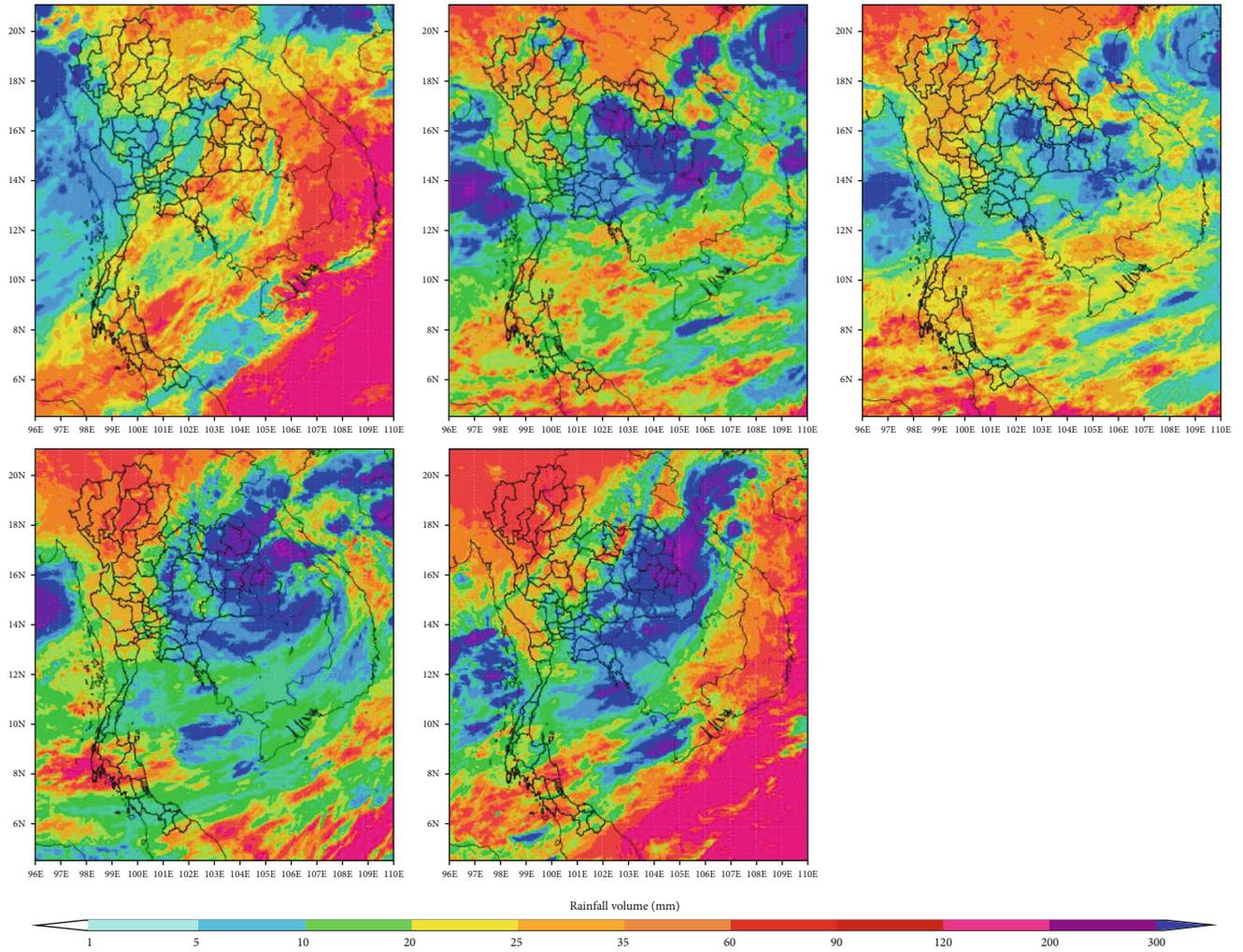
The normalized temperature ( $N_{\text{temp}_c}$ ) can be expressed as

$$N_{\text{temp}_c} = \begin{cases} \text{zero}; & \text{temp}_k - \theta > \varepsilon, \\ \text{temp}_k^{0.5}; & \text{temp}_k - \theta < \varepsilon, \end{cases} \quad (2)$$

where  $\theta = 273$  and  $\varepsilon$  is a parameter using the color bar monitoring.

Equation (2) is to estimate the amount of precipitation fallen in the form of temperature with the satellite data across the region. To convert to the rainfall rate, the results from Equation (2) are used in Equation (1) to calculate with the specific coefficients parameter referred to the climates and locations.

The proposed fast satellite data processing (Fast-SDP) algorithm is introduced in Algorithm 1. Line 1-2, the initial satellite data and 9 types of network common satellite data forms, called “.nc files”, are input to the host processor and converted to the common data form language called a “.cdl file” prepared for the next processing. The files are



(a) Satellite-based estimated rainfall of tropical storm *Kajigi* before qualitative precipitation estimation (QPE)

FIGURE 4: Continued.

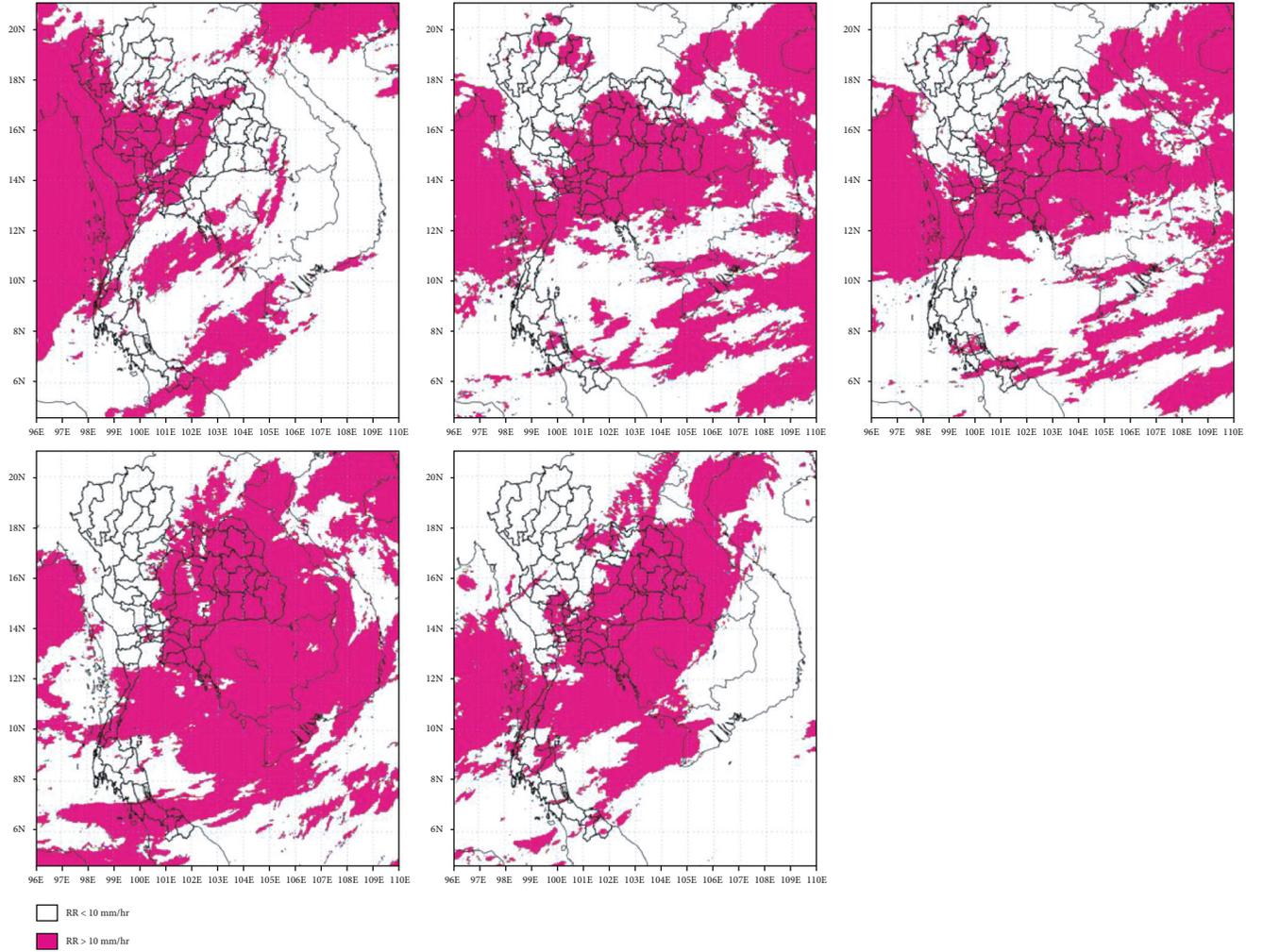
(b) Satellite-based estimated rainfall of tropical storm *Kajigi* after QPEFIGURE 4: Satellite imageries of tropical storm *Kajigi* between 31 August and 4 September 2019, processed by the proposed Fast-SDP algorithm.

TABLE 3: Performance evaluation of the proposed algorithm in different platforms.

Programming model	Algorithm	API/language	SDK	HW platform	Processing time (sec)	Overhead time (sec)	Overall (sec)
Distributed parallel	Fast-SDP	OpenMP & MPI & ESDK & C/C++	ESDK	DDSA	91.15	24.38	115.53
	SDP [31]	MPI & C/C++	IDE	DDSA	69.48	1407.96	1477.44
	SDP [30]	MPI & ESDK & C/C++	ESDK	DDSA	925.65	N/A	925.65
Parallel	Fast-SDP	C/C++	IDE	DSA	682.56	99.27	781.83
	SDP [30]	C/C++	IDE	DSA	826.56	105.12	931.68
			Xcode	iMac	45.36	N/A	45.36
	SDP [32]	C/C++	IDE	MacBook	39.78	5.85	45.63

distributed to the 8 sDSAs through the mDSA using the OpenMP API command. Lines 3 and 4 show the parallelism of OpenMP, searching for  $temp_k$  in the .cdl file at the coprocessor. While running the algorithm, the OpenMP API exploits several threads to search  $temp_k$  in the nine .cdl files

as a parallel process. Line 5–7, the decision on the rain rate is made. The results are returned to the host processor when finished. Line 8 shows the conversion process of the .cdl files back to the .nc file. Line 9, the satellite data processing of all .nc files are collected at the host processor.

TABLE 4: Comparisons of the results from proposed Fast SDP after QPE with the observation data [33].

	Correlation coefficient		Euclidean distance	
	Himawari-8 satellite (before QPE) and fast-SDP results (after QPE)	Fast-SDP results (after QPE) and real observation data [33]	Himawari-8 satellite (before QPE) and fast-SDP results (after QPE)	Fast-SDP results (after QPE) and real observation data [33]
Podul:				
27/08/2019	0.3163	0.9924	0.4160	0.0103
28/08/2019	0.3176	0.9955	0.4361	0.0039
29/08/2019	0.6168	0.8760	0.5362	0.1275
30/08/2019	0.6258	0.8473	0.5362	0.1226
31/08/2019	0.3780	0.9868	0.4154	0.0309
Kajigi:				
31/08/2019	0.4021	0.9868	0.4085	0.0305
1/09/2019	0.4398	0.9659	0.5337	0.0608
2/09/2019	0.3852	0.9739	0.5048	0.0519
3/09/2019	0.5409	0.9484	0.5351	0.0807
4/09/2019	0.4462	0.9862	0.3987	0.0294

## 5. The Experimental Results

Experiments on the proposed DDSA concerning QPE from Himawari-8 satellite data are performed based on the Fast-SDP algorithm. We implement the proposed Fast-SDP algorithm in Epiphany-C/C++, compiled by E-GCC/G++ on our DDSA. The Fast-SDP algorithm is used to optimize the QPE process, including receiving and managing the raw data of the movement of tropical storms *Podul* and *Kajigi* over Thailand during 27-31 August 2019 and 31 August to 4 September 2019 from the Himawari-8 satellite, available in every 10 minutes every day, to estimate the rainfall from the rain-temperature relationship. Initial parameters for rain-temperature calculation followed the climates and locations in Thailand [29, 30]  $\phi = 1.1183$ ,  $\psi = -0.036382$ ,  $\zeta = 0.5$ , and  $\varepsilon = -25$ .

Figures 3 and 4 show the satellite imageries generated by the Fast-SDP algorithm for Thailand during 27-31 August 2019 for the tropical storm *Podul* and during 31 August to 4 September 2019 for the tropical storm *Kajigi* where high temperature is represented in red color, while low temperature is in blue color. The shades of colors were translated using Equation (1), followed by a software called GrADS (grid analysis and display system) for graphical display in a default rainbow color code.

The performance comparison of the proposed Fast-SDP algorithm implemented in different platforms is shown in Table 3. Mainly, there are two types of programming in this evaluation, (1) distributed parallel programming in DDSA hardware platforms consisting of one master and 8 slaves as proposed in section 3.1 and (2) parallel programming using coprocessors in a Parallella DSA board or in PC (i.e., iMac and MacBook).

As can be seen, the proposed Fast-SDP runs faster than the SDP in [30, 31] in the DDSA hardware platform. The processing time can be reduced from 1477 s [31] and 925 s [30] to 115 s, or we can, respectively, achieve 12 and 8 times faster. For a single CPU with multicoprocessor platform,

the proposed algorithm can run in parallel using its local coprocessors. Although running in desktops or notebooks is faster than running in the embedded DSA board, power consumption is higher, also the price.

The estimated rainfall rate using the Fast-SDP algorithm shown in Figures 3(b) and 4(b) is verified by comparing with the actual observation data from the Thai Meteorological Department (TMD) provided in [33]. The correlation coefficient and Euclidean distance techniques are employed for this comparison.

The correlation coefficient [34, 35] is used to compare the histograms of the results image in Figures 3(b) and 4(b) and the existing measures data from [33]. The correlation coefficient  $\mathfrak{D}(\mathcal{H}_1, \mathcal{H}_2)$  can be expressed to match two histograms  $\mathcal{H}_1$  and  $\mathcal{H}_2$  by

$$\mathfrak{D}(\mathcal{H}_1, \mathcal{H}_2) = \frac{\sum_{I=1}^N (\mathcal{H}_1(I) - \overline{\mathcal{H}_1}) (\mathcal{H}_2(I) - \overline{\mathcal{H}_2})}{\sqrt{\sum_{I=1}^N (\mathcal{H}_1(I) - \overline{\mathcal{H}_1})^2 \cdot \sum_{I=1}^N (\mathcal{H}_2(I) - \overline{\mathcal{H}_2})^2}}, \quad (3)$$

where

$$\overline{\mathcal{H}_\alpha} = \frac{1}{\mathcal{N}} \sum_{I=1}^N \mathcal{H}_\alpha(I), \quad \alpha = 1, 2, \quad (4)$$

and  $\mathcal{N}$  is the total number of histogram bins,  $\mathcal{H}_1$  is from the results images in Figures 3(b) and 4(b), and  $\mathcal{H}_2$  is from the actual observation data from [33]. The property of correlation coefficient is in between  $0 \leq \mathfrak{D}(\mathcal{H}_1, \mathcal{H}_2) \leq 1$ . The value of correlation coefficient is close to 1, which means that the two histograms  $\mathcal{H}_1$  and  $\mathcal{H}_2$  have strong correlations.

The Euclidean distance method [36] is to calculate by the Euclidean plane. We define that the point  $\mathcal{P}$  has Cartesian coordinate  $(\mathcal{P}_\eta, \mathcal{P}_\zeta)$  and the point  $\mathcal{Q}$  has coordinate  $(\mathcal{Q}_\eta, \mathcal{Q}_\zeta)$ . So, the pairwise distance  $\mathbb{D}(\mathcal{P}, \mathcal{Q})$  between the points

of  $\mathcal{P}_\eta$  and  $\mathcal{Q}_\eta$  is given by

$$\mathbb{D}(\mathcal{P}, \mathcal{Q}) = \sqrt{(\mathcal{P}_\eta - \mathcal{P}_\zeta)^2 + (\mathcal{Q}_\eta - \mathcal{Q}_\zeta)^2}. \quad (5)$$

If the distance is closer to zero, it means that the two points can simply be defined as the similar points.

Table 4 shows the correlation coefficient and the Euclidean distance between two sets of the data, (a) Himawari-8 Satellite before QPE & Fast-SDP results after QPE and (b) the Fast-SDP results and real observation data from [33]. The average correlation coefficient from the Fast-SDP results after QPE compared with the real observation data of *Podul* storm and *Kajigi* is about 0.9396 and 0.9722, respectively. The average of Euclidean distance from the Fast-SDP results after QPE compared with the real observation data of *Podul* storm and *Kajigi* is about 0.0590 and 0.0507, respectively. This confirms the similarity of the results from the proposed algorithm and the real observation data.

## 6. Conclusion

In this paper, the fast satellite data processing (Fast-SDP) algorithm was proposed for QPE using Himawari satellite data and based on a distributed domain-specific architecture (DDSA). The proposed master-slave architecture consists of 1 mDSA and 8 sDSAs connected by a gigabit switch. The Fast-SDP algorithm is receiving and managing raw data from the satellite every 10 minutes and calculating the rain temperature. The experimental results of satellite imageries can be used to visualize tropical storm movement relating to the rainfall over Thailand. From the performance evaluation results, the overall QPE processing time using the proposed Fast-SDP is 115.53 seconds or 8 times faster than the previous work [30]. Additionally, the correctness of the results from the proposed Fast-SDP after QPE is confirmed by the real observation data from the Thai Meteorological Department (TMD) using correlation coefficient and Euclidean distance.

The novelty of this study is the proposed Fast-SDP for solving QPE in the distributed domain-specific architecture, using data from the Himawari satellite. This work provides advantages for the TMD in terms of network flexibility, hardware scalability, low power consumption, and suitable computational performance for 10-minute window of incoming satellite data.

## Data Availability

Real observation data is supported by the Thai Meteorological Department, Thailand.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

We would like to thank Dr. Kamol Promasakha na Sakolnakhon and Mr. Peeranat Longsombun at the Meteorological Radar and Satellite Data Analysis Division, The Meteorological Department in Bangkok, Thailand, and Mr. Jirayut Omkhet and Mr. Thanathip Phikulthong at the Department of Computer Engineering, Faculty of Industrial Technology, Suan Sunandha Rajabhat University in Bangkok, Thailand.

## References

- [1] H. Chen, R. Yu, and Y. Shen, "A new method to compare hourly rainfall between station observations and satellite products over central-eastern China," *Journal of Meteorological Research*, vol. 30, no. 5, pp. 737–757, 2016.
- [2] J. Panda, H. Singh, P. K. Wang, R. K. Giri, and A. Routray, "A qualitative study of some meteorological features during tropical cyclone PHET using satellite observations and WRF modeling system," *Journal of Indian Society of Remote Sensing*, vol. 43, no. 1, pp. 45–56, 2015.
- [3] R. F. Adler, G. Gu, M. Sapiano, J. Wang, and G. J. Huffman, "Global precipitation: means, variations and trends during the satellite era (1979–2014)," *Surveys in Geophysics*, vol. 38, no. 4, pp. 679–699, 2017.
- [4] V. N. Malinin and S. M. Gordeeva, "Variability of evaporation and precipitation over the ocean from satellite data," *Atmospheric and Oceanic Physics*, vol. 53, no. 9, pp. 934–944, 2017.
- [5] Y. Li, X. Pei, Q. Huang, L. Jiao, R. Shang, and N. Marturi, "Anchor-free single stage detector in remote sensing images based on multiscale dense path aggregation feature pyramid network," *IEEE Access*, vol. 8, pp. 63121–63133, 2020.
- [6] M. Zhang, Y. Chen, X. Liu, B. Lv, and J. Wang, "Adaptive anchor networks for multi-scale object detection in remote sensing images," *IEEE Access*, vol. 8, pp. 57552–57565, 2020.
- [7] Y. Li and D. Ye, "Greedy annotation of remote sensing image scenes based on automatic aggregation via hierarchical similarity diffusion," *IEEE Access*, vol. 6, pp. 57376–57388, 2018.
- [8] R. Yu, M. Lyu, J. Lu, Y. Yang, G. Shen, and F. Li, "Spatial coordinates correction based on multi-sensor low-altitude remote sensing image registration for monitoring forest dynamics," *IEEE Access*, vol. 8, pp. 18483–18496, 2020.
- [9] J. L. Hennessy and D. A. Patterson, *Computer Architecture, Sixth Edition: A Quantitative Approach*, Morgan Kaufmann, Cambridge, MA, USA, 2019.
- [10] S. Prongnuch and T. Wiangtong, "Heterogeneous computing platform for data processing," in *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–4, Phuket, Thailand, 2016.
- [11] Mahanakorn University of Technology, S. Prongnuch, S. Sitjongsataporn, Mahanakorn University of Technology, T. Wiangtong, and King Mongkut's Institute of Technology Ladkrabang, "A heuristic approach for scheduling in heterogeneous distributed embedded systems," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 1, pp. 135–145, 2020.
- [12] Thailand Meteorological Department, *Vision and Missions* August 2019, <https://www.tmd.go.th/en/aboutus/vision.php>.
- [13] *HPE APOLLO 2000 GEN10 SYSTEM: The Enterprise Bridge to Scale-out Architecture* September 2020, <https://h20195.www2.hp.com/v2/GetPDF.aspx/4AA4-8164ENW.pdf>.

- [14] B. Archevarahuprok, *TMD's Numerical Weather Prediction System*, 2020.
- [15] S. Hong, D. Shin, B. Park, and D. So, "Development of prototype algorithms for quantitative precipitation nowcasts from AMI onboard the GEO-KOMPSAT-2A satellite," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7149–7156, 2016.
- [16] V. Petkovic, C. D. Kummerow, D. L. Randel, J. R. Pierce, and J. K. Kodros, "Improving the quality of heavy precipitation estimates from satellite passive microwave rainfall retrievals," *Journal of Hydrometeorology*, vol. 9, pp. 69–85, 2018.
- [17] G. Xie, G. Zeng, X. Xiao, R. Li, and K. Li, "Energy-efficient scheduling algorithms for real-time parallel applications on heterogeneous distributed embedded systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3426–3442, 2017.
- [18] H. Jia and N. Verma, "Exploiting approximate feature extraction via genetic programming for hardware acceleration in a heterogeneous microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 4, pp. 1016–1027, 2018.
- [19] S. J. Matthews, R. W. Blaine, and A. F. Brantly, "Evaluating single board computer clusters for cyber operations," in *2016 International Conference on Cyber Conflict (CyCon U.S.)*, pp. 1–8, Washington, DC, USA, October 2016.
- [20] F. Gao, Z. Huang, S. Wang, and X. Ji, "A scalable object detection framework based on embedded manycore cluster," in *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 142–145, Chengdu, China, 2016.
- [21] C. Chen, K. Li, A. Ouyang, Z. Zeng, and K. Li, "GfLink: an in-memory computing architecture on heterogeneous CPU-GPU clusters for big data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1275–1288, 2018.
- [22] Adapteva Inc, *Parallella-1.x reference manual* August 2019, <https://www.parallella.org/docs/parallella-manual.pdf>.
- [23] O. Jeppsson, *Parabuntu 2019.1 beta1* May 2020, [https://www.parallella.org/docs/parallella\\_manual.pdf](https://www.parallella.org/docs/parallella_manual.pdf).
- [24] Brown Deer Technology, LLC. *COPRTHR-2* May 2020 <https://www.browndeertechnology.com/sw-coprthr.html>.
- [25] *Mathematics and Computer Division Argonne National Laboratory. MPICH user's guide* May 2020, <https://www.mpich.org/static/downloads/3.3.2/mpich-3.3.2-userguide.pdf>.
- [26] *The OpenMP API specification for parallel programming* June 2020, <https://www.openmp.org/about/about-us/>.
- [27] *Beyond Linux From Scratch. NFS-Uutils-2.4.3* May 2020, <http://www.linuxfromscratch.org/blfs/view/svn/basicnet/nfs-utils.html>.
- [28] N. I. C. T. ScienceCloud, *Himawari real-time* June 2020, <https://himawari8.nict.go.jp/>.
- [29] K. P. N. Sakolnakhon and S. Nuntakamolwaree, "The estimation rainfall using infrared (IR) band of Himawari-8 satellite over Thailand," *Engineering: Naresuan University*, vol. 39, pp. 236–248, 2016.
- [30] S. Prongnuch, S. Sitjongsataporn, and T. Wiangtong, "Performance evaluation of heterogeneous cluster for satellite data processing," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–2, Beitou, Taiwan, 2019.
- [31] S. Prongnuch, J. Omkhet, T. Phikulthong, and P. Malapim, "The mobile high-performance computing design for satellite data processing (in Thai)," in *Proc. of RSU National Research Conference on Science and Technology, Social Sciences and Humanities*, pp. 298–307, Rangsit University, Thailand, 2019.
- [32] S. Prongnuch, "SoC satellite data processing for tropical storm Podul in Thailand (in Thai)," in *Proceedings of ECTI-CARD*, pp. 52–56, Nakhon Sawan, Thailand, 2020.
- [33] Thailand Meteorological Department, *Analysis of satellite images (in Thai)* January 2021, <http://www.satda.tmd.go.th/>.
- [34] R. C. Gonzalez, *Digital Image Processing*, Pearson, 2008.
- [35] OpenCV, *Histogram comparison* Jan. 2021, [https://docs.opencv.org/3.4/d8/dc8/tutorial\\_histogram\\_comparison.html](https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html).
- [36] A. Y. Alfakih, *Euclidean distance matrices and their applications in rigidity theory*, Springer, 2018.