

Communication

# Performance Evaluation of Parallel Structure from Motion (SfM) Processing with Public Cloud Computing and an On-Premise Cluster System for UAS Images in Agriculture

Anjin Chang <sup>1,\*</sup>, Jinha Jung <sup>2</sup>, Jose Landivar <sup>3</sup>, Juan Landivar <sup>3</sup>, Bryan Barker <sup>4</sup> and Rajib Ghosh <sup>4</sup>

<sup>1</sup> School of Engineering and Computing Sciences, Texas A&M University-Corpus Christi, Corpus Christi, TX 78412, USA

<sup>2</sup> Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA; jinha@purdue.edu

<sup>3</sup> Texas A&M AgriLife Research & Extension Center at Corpus Christi, Corpus Christi, TX 78406, USA; jose.landivarscott@ag.tamu.edu (J.L.); jalandivar@ag.tamu.edu (J.L.)

<sup>4</sup> Oracle for Research Program, Austin, TX 78741, USA; bryan.barker@oracle.com (B.B.); rajib.ghosh@oracle.com (R.G.)

\* Correspondence: anjin.chang@tamucc.edu; Tel.: +1-361-825-3165

**Citation:** Chang, A.; Jung, J.; Landivar, J.; Landivar, J.; Barker, B.; Ghosh, R. Performance Evaluation of Parallel Structure from Motion (SfM) Processing with Public Cloud Computing and an On-Premise Cluster System for UAS Images in Agriculture. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 677. <https://doi.org/10.3390/ijgi10100677>

Academic Editors: José R.R. Viqueira, José M. Cotos and Aurora Cuartero

Received: 28 August 2021

Accepted: 4 October 2021

Published: 7 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

**Abstract:** Thanks to sensor developments, unmanned aircraft system (UAS) are the most promising modern technologies used to collect imagery datasets that can be utilized to develop agricultural applications in these days. UAS imagery datasets can grow exponentially due to the ultrafine spatial and high temporal resolution capabilities of UAS and sensors. One of the main obstacles to processing UAS data is the intensive computational resource requirements. The structure from motion (SfM) is the most popular algorithm to generate 3D point clouds, orthomosaic images, and digital elevation models (DEMs) in agricultural applications. Recently, the SfM algorithm has been implemented in parallel computing to process big UAS data faster for certain applications. This study evaluated the performance of parallel SfM processing on public cloud computing and on-premise cluster systems. The UAS datasets collected over cropping fields were used for performance evaluation. We used multiple computing nodes and centralized network storage with different network environments for the SfM workflow. In single-node processing, an instance with the most computing power in the cloud computing system performed approximately 20 and 35 percent faster than in the most powerful machine in the on-premises cluster. The parallel processing results showed that the cloud-based system performed better in speed-up and efficiency metrics for scalability, although the absolute processing time was faster in the on-premise cluster. The experimental results also showed that the public cloud computing system could be a good alternative computing environment in UAS data processing for agricultural applications.

**Keywords:** UAS; Structure from Motion (SfM); cloud computing

## 1. Introduction

In recent days, unmanned aircraft system (UAS) have been actively utilized in agricultural applications to develop a high-throughput phenotyping (HTP) system [1,2]. UAS, often called as drone, can collect high-spatiotemporal-resolution imagery data over agricultural fields. UAS data can be processed to visualize agriculture fields and analyzed for developing advanced agriculture applications [3]. Once UAS data are collected in the field, the data need to be processed to extract phenotypic information. The structure from motion (SfM) algorithm is the most popular algorithm used to turn numerous UAS images with significant overlaps into measurable geospatial data products such as 3D point clouds, digital elevation models (DEMs), and orthomosaic images using the triangulation

concept in photogrammetry. The geospatial data products generated from the SfM process are then adopted to generate georeferenced phenotypic information [4–7].

As hundreds of images are easily taken by each UAS flight mission, UAS data collection campaigns usually result in a huge imagery dataset. Although computing power has rapidly increased recently, processing massive amounts of UAS data is still a challenging task, as the computational resource requirements grow exponentially as the number of UAS images increases [8,9]. The SfM process could take many hours or even days to process big UAS data collected at a fine spatiotemporal resolution. To overcome this hurdle, high-performance computing (HPC) capabilities can be adopted to parallelize the SfM process and expedite the computation time.

For the parallel processing of SfM, a cluster system with independent computers and common storage can be employed. Although cluster systems have benefits such as high performances, fault tolerance, and scalability, users should invest significant resources including labor, hardware, and software to construct and maintain the system locally [10]. In recent years, since commercial cloud computing service have developed rapidly and inexpensively, cloud computing systems can serve as an effective counterplan for cluster computing.

To evaluate the potential for employing cloud computing systems to process UAS data for agriculture fields, the performance of SfM processing must be examined. Therefore, two UAS datasets in two different environments were tested in public cloud computing and on-premises cluster systems. The main objective of this study was to: (1) compare the performance of single-node processing with different computing power and storage options and (2) test parallel processing in public cloud-based and on-premises cluster systems. For the experiments, the high-quality RGB imagery was collected by using a UAS platform, and then processed with SfM software in various environments. The processing time was measured and used to compare the performance of the cloud-based and on-premises cluster systems.

## 2. Materials and Methods

### 2.1. UAS Datasets

In this study, two UAS missions were designed to collect small (2.3 GB over 4 acre) and large (12.5 GB over 220 acre) datasets to conduct performance comparison tests. A field for a small dataset was located in the research farm managed by Texas A&M AgriLife Research and Extension Center at Corpus Christi. Corn crops were planted in this field. The large dataset covered cotton and sorghum plants in a commercial field in Driscoll, TX. RGB images were collected with a DJI Phantom 4 RTK (DJI, Shenzhen, China) for both fields. The onboard camera (FC6310R) was equipped with a 20 megapixel CMOS sensor, a resolution of  $5,472 \times 3,648$ , an 8.8-mm focal length, and an  $84^\circ$  field of view (FOV). The flight parameters, such as the flight altitude and overlap, were determined by the field size and flight time (Table 1). One and four UAS flights were conducted to collect 293 and 1,557 images for small and large datasets, respectively. As we used the same UAS platform and sensor, the total volume of raw images was directly proportional to the number of images.

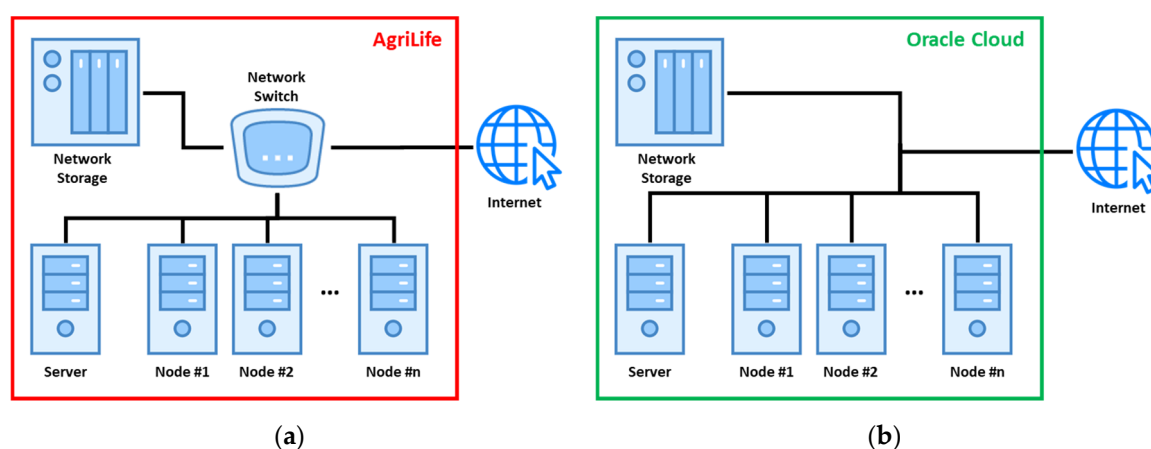
**Table 1.** Details of small and large datasets.

	Small Dataset	Large Dataset
Acquisition date	2020/05/19	2020/04/30
Field size (acre)	4	220
Flight altitude (m)	25	90
Overlap (%)	85	70
# of Images	293	1557
Total data size (GB)	2.3	12.5
GSD (cm)	0.7	2.5

## 2.2. Cluster Systems for Processing

Two cluster systems were adopted to test performance of processing UAS images with the SfM algorithm. The AgriLife Local Cluster (ALC) is an isolated on-premises cluster constructed in the Texas A&M AgriLife Research & Extension Center at Corpus Christi. The ALC consists of 5 workstations (nodes) and network-attached storage (NAS). All nodes and the NAS are interconnected through a network switch with gigabit ethernet ports (Figure 1a). NAS was connected to the network switch with four gigabit LAN ports to increase the bandwidth by aggregating multiple network interfaces and preventing traffic failover to maintain network connections. All nodes and NAS could communicate internally regardless of public internet connections. Each machine has different hardware such as CPU, RAM, and graphic card, which are equipped for each node in the ALC (Table 2). The details of the CPU and GPU specifications are shown in Appendix A (Tables A1 and A2).

An oracle cloud cluster (OCC) was built with various combinations of four compute shapes and two storage options in the oracle cloud infrastructure (OCI) (Table 2). Although all shapes employ the same CPU, the numbers of CPUs, CPU/GPU memory, and network bandwidth are different for each shape (Table A3) [11]. Two storage options, file storage and block volume, were tested in this study. The block volume was used as the local storage of the node, while file storage worked as a network drive in OCC [12,13]. For a multi-nodes cluster system, nodes and network storage were set up in the OCI and connected through the public internet (Figure 1b). The OCC was simple and easy to build in the OCI for parallel processing, but the network speed through the public internet mainly affected the processing time.



**Figure 1.** Cluster architecture of (a) the ALC and (b) the OCC. Nodes and network storage can be communicated through network switching internally without the internet in ALC. All components in the OCC had their own IP address (IPv4) to use the public internet for communication.

**Table 2.** Summary of Hardware Specification of nodes in AgriLife Cluster and OCI.

System	Node	OS	Processor	RAM	GPU (#)
AgriLife Cluster	M1		Intel(R) Core(TM) i7-8700K	32 GB	GeForce GTX 1070 Ti
	M2		Intel(R) Core(TM) i7-4790K	32 GB	GeForce GTX 980
	M3	Windows 10 (Build 20H2)	Intel(R) Xeon(R) E5-2680	64 GB	GeForce GTX 1050 Ti
	M4		Intel(R) Xeon(R) E5-1650	32 GB	GeForce GTX 1050 Ti
	M5		Intel(R) Xeon(R) E5-1650	32 GB	GeForce GTX 1050 Ti

Oracle Cloud	VM.GPU2.1	Windows Server 2019	Intel(R) Xeon(R) Platinum 8167 M	72 GB	NVIDIA Tesla P100 (×1)
	VM.GPU3.1		Intel(R) Xeon(R) Platinum 8167 M	90 GB	NVIDIA Tesla V100 (×1)
	VM.GPU3.2		Intel(R) Xeon(R) Platinum 8167 M	180 GB	NVIDIA Tesla V100 (×2)
	VM.GPU3.4		Intel(R) Xeon(R) P latinum 8167 M	360 GB	NVIDIA Tesla V100 (×4)

### 2.3. Structure from Motion (SfM) Processing

Although there are various available SfM software programs, such as Agisoft Metashape, Pix4D, and OpenDroneMap, and image mosaicking services by DroneDeploy, Agisoft Metashape software (1.6.3.10732, 64 bit) is used to process UAS raw images. Agisoft Metashape also provides network (parallel) processing using multiple nodes as well as stand-alone processing. In this study, Agisoft Metashape was selected to process UAS data through batch processing to avoid manual work in the processing pipeline (Table 3). Although Metashape provides many parameters that the user can adjust, default or recommended options were used for all of the experiments.

In the align photos, Metashape estimated the camera position at the time of image capture defined by the interior and exterior orientation parameters [14]. Interior orientation (IO) parameters included the camera focal length, coordinates of the image principal point and lens distortion coefficients. Exterior orientation (EO) parameters defined the position and orientation of the images. EO consisted of 3 translation components (X, Y, and Z) and 3 Euler rotation angles (yaw, roll, and pitch). The UAS platform used in this study is equipped with RTK GPS systems for measuring initial EO parameters in image capture. IO and EO parameters can be calculated by Metashape using aerotriangulation with tie points and bundle block adjustment based on collinearity equations [15]. After this processing, estimated IO and EO with sparse point cloud containing triangulated positions of matched image points were resulted.

A depth map calculated using dense stereo image matching is constructed for the overlapping image pairs considering the updated IO and EO parameters from the previous process. In Metashape, the depth map is transformed into partial dense point clouds, and then it is merged into a final dense point cloud. For every point in the final dense point cloud, a confidence value, which means the number of contributing depth maps, and color information sampled from the images are stored.

In this study, the DEM is rasterized from a dense point cloud with height values stored per every cell on the regular grid, and then used to build the orthomosaic. A combined image created by the seamless merging of the raw images was projected on the ground surface with the selected projection. As file saving is conducted by a single-node, DEM and orthomosaic images are also exported to compare the performance of different storage options.

**Table 3.** SfM processing pipeline and options.

Procedure	Default Values
Align Photos	Accuracy: High, Key point limit: 40,000, tie point limit: 4000, Adaptive camera model fitting: Yes
Build Dense Cloud	Quality: High, Filtering mode: Mild, Calculate point cloud: Yes
Build DEM	Source data: Dense cloud, Interpolation: Enabled
Build Orthomosaic	Blending mode: Mosaic, Surface: DEM, Enable hole filling: Yes

Export DEM	File format: GeoTIFF, Pixel size: Default, Write tiled TIFF: Yes, Write BigTIFF file: Yes, Generate TIFF overview: Yes
Export Orthomosaic	File format: GeoTIFF, Pixel size: Default, Write tiled TIFF: Yes, Write BigTIFF file: Yes, Generate TIFF overview: Yes, Write alpha channel: Yes

#### 2.4. Performance Testing

The performance experiments were conducted using single and multiple nodes for the small and large datasets. In single-node processing, two different storage environments were tested in the different cluster environment. For local and network storage options, all UAS data and processing products were stored in the local hard drive and network drive in both cluster systems. Due to the speed of the disk I/O (input and output) and network, local storage could be expected to process faster. Three workstations (M1, M2, and M3) in the ALC and four VMs (2.1, 3.1, 3.2, and 3.4) in the OCC were selected for single-node processing to compare the performance with different computing powers in a single-node. For multi-nodes processing, the datasets were processed using the network processing mode in Metashape. The processing began with one node, and then additional nodes were used, up to five and six nodes in the ALC and OCC, respectively.

All processes were conducted continuously in a batch process without manual work. Processing time was measured as a criterion of performance. All experiments were repeated three times and the average processing time was used for comparison.

The speed-up and efficiency, the principal measurements of parallelization efficiency, were calculated from the total computation time in multi-nodes processing. The Speed-Up ( $S_N$ ) was defined as the ratio of the time required to execute the computational workload on a single-node to the time required for the same task on N processors [16]:

$$S_N = \frac{T_1}{T_N}, \quad (1)$$

where  $T_1$  is the execution time on a single processor and  $T_N$  is the execution time on N processors.

Efficiency was defined as the ratio of speed-up to the number of processors (Equation (2)) [17]:

$$E_N = \frac{S_N}{N}, \quad (2)$$

where  $E_N$  is the efficiency on N processors,  $S_N$  is the speed-up on N processors, and N is the number of processors. Efficiency can be used to measure the fraction of time for which each node is usefully utilized.

### 3. Results

#### 3.1. Computing Power of Single-Node

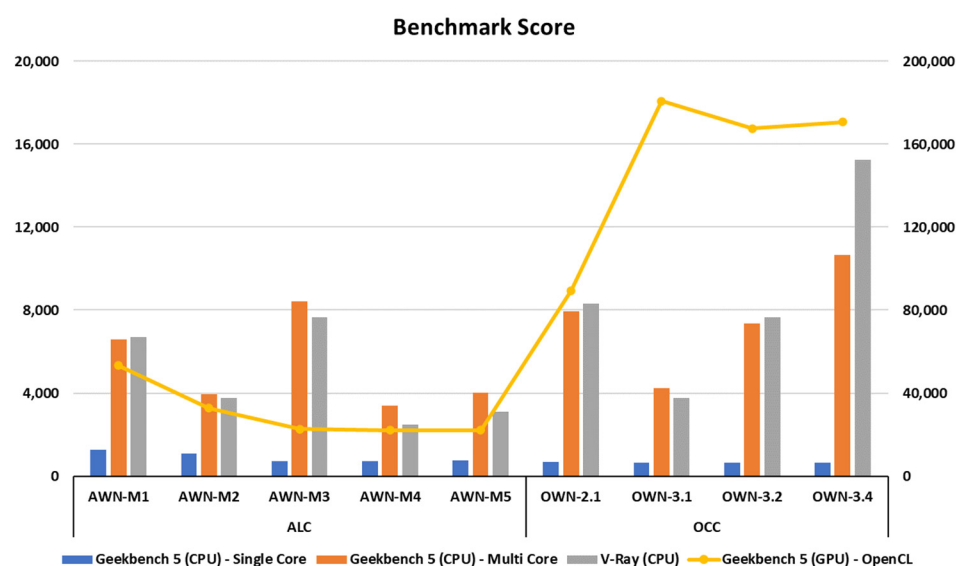
To show the computation power of each node in the ALC and OCC, the benchmark scores were measured in different environments. In Figure 2, items on the X-axis indicate the abbreviations of each node. The first letter means the ALC (A) or OCC (O), and the third letter means local (L) or network (N) storage. Second term is showing the machine ID (ex. M1, M2, etc.) in the ALC or the shape (ex. 2.1, 3.1, etc.) in the OCC. Single-core and multi-core power indicating the overall performance of main processor was measured by GeekBench 5 and V-Ray. GPU performance was also tested by GeekBench 5 (OpenCL).

In the ALC, all nodes showed different performances because each node was equipped with different hardware specifications. Although the single-core powers of M1 and M2 were higher than those of the others, M3 was the highest in the multi-core test because it had the largest number of cores (threads). GPU power was strongly related to

the specification of the graphic card. M1 was the highest, while M3, M4, and M5 were similar in the GPU test.

The single-core power of the nodes in OCC was lower than M1 and M2 due to the frequency of the CPU, but the multi-core power of OCC was higher. Though VM.GPU2.1 consisted of more CPU/memory and faster network bandwidth, VM.GPU2.1 showed better performance of multi-core power than VM.GPU3.1, but similar to VM.GPU3.2, which was equipped with the same number of OCPU. The VM.GPU3 series equipped a more powerful GPU than two times of VM.GPU2.1 and four times all nodes in the ALC.

Based on the benchmark test applied to each single-node in the ALC and OCC, we tested which hardware parts were more influential in SfM processing and the potential of cloud-based clusters for UAS data processing.



**Figure 2.** Results of the benchmark test using GeekBench 5 and V-Ray to measure the power of the CPU and GPU for each single-node in the ALC and OCC. The higher scores indicate better performance.

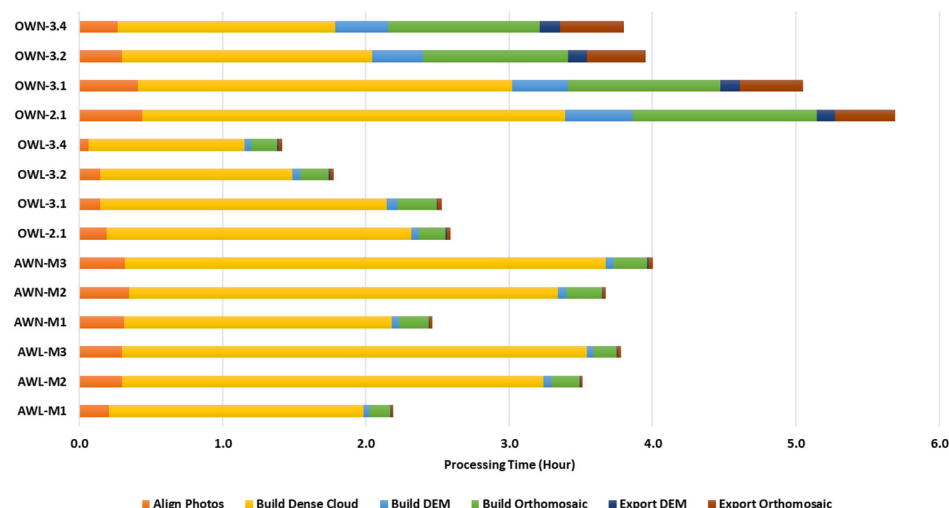
### 3.2. Single-Node Processing

The performance for SfM processing in single-node was tested in experiments based on: (1) hardware specifications; (2) storage options; (3) and the UAS data size (Figure 3). In the same cluster system, the node with the more powerful GPU performed better in processing UAS data. For example, AWN/AWL-M1 was approximately 40 percent faster than AWN/AWL-M3 when using the small dataset, even though M3 resulted in a higher score on the multi-core benchmark. The OCC, VM.GPU3 series was also faster than VM.GPU2.1 when using the small dataset. Moreover, the results of large dataset processing showed that multi-core capability is another factor of processing speed. For example, GeekBench 5 and V-Ray showed a linear increase with the number of GPU cores for VM.GPU3 series shapes (Figure 2). Despite of the higher GPU performance, AWL/AWN-M2 and OWL/OWN-3.1 were slower than the other nodes when using the large dataset due to the power of the CPU. This implies that GPU and multi-core factors are highly influential hardware specifications in single-node processing.

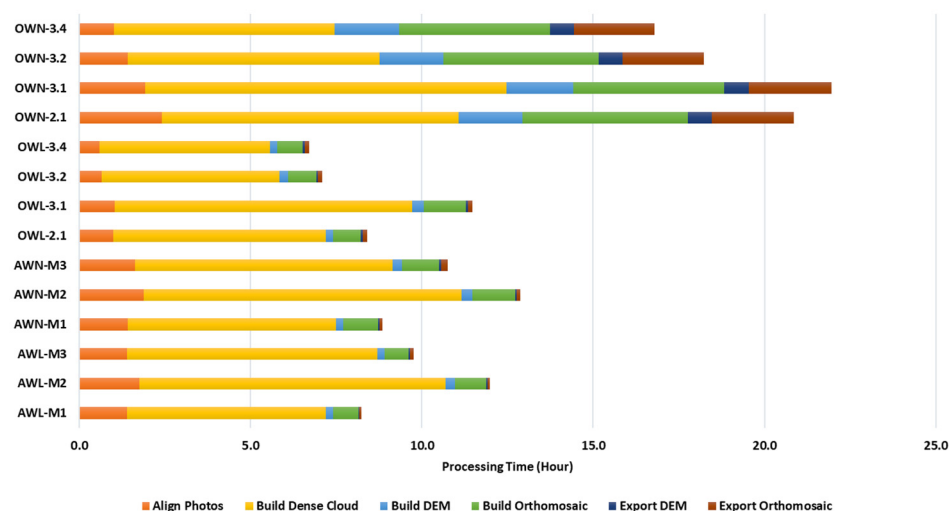
In single-node processing, performance time was critically affected by different storage options, especially, in the OCC. Local storage, which is a block storage, is generally faster than network storage. In the ALC, there was less than a 10 percent difference between local and network storage, since the network speed through the router was as fast as local disk I/O. However, the network storage (file storage option) in OCC made SfM processing twice as slow. As Metashape must communicate with the storage through the entire processing, the disk I/O speed mainly affected the processing time of each step. The

disk I/O-intensive works, such as Build Dem, Build Orthomosaic, and Export DEM/Orthomosaic took significantly longer time with network storage in OCC. These processes occupied approximately 45~55% of the entire processing time.

The results of the comparison between the ALC and OCC in single-node processing demonstrated that a cloud computing system could provide more performance gain with the appropriate virtual machine shape and storage architecture. For example, OWL-3.4 performed approximately 20 and 35 percent faster than AWL-M1, which is the fastest machine in the ALC, for the large and small datasets, respectively.



(a)



(b)

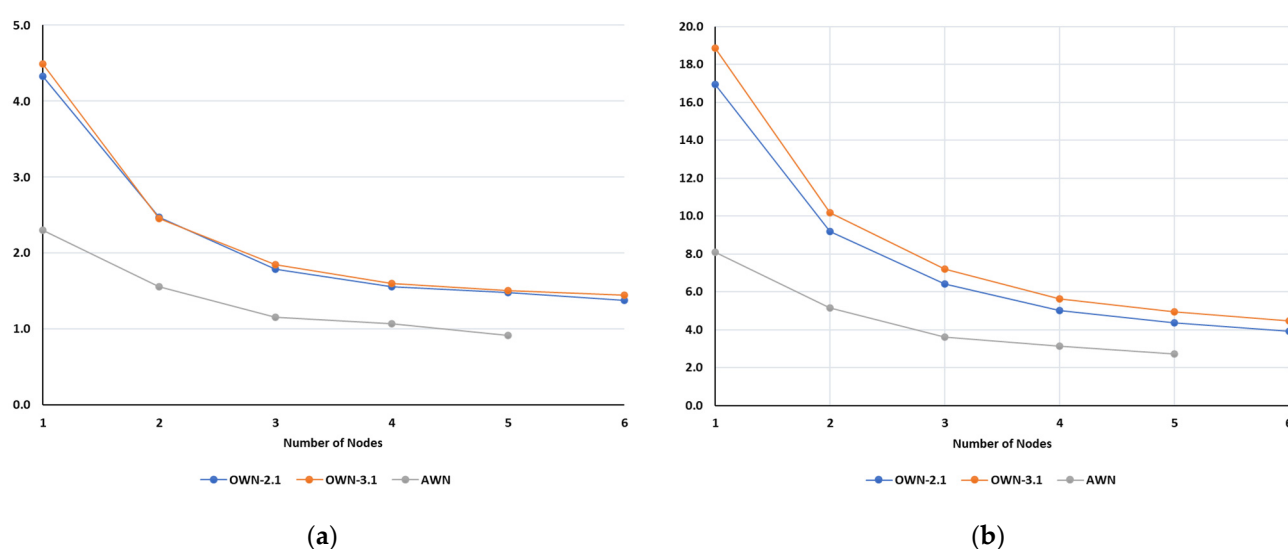
**Figure 3.** Processing time of SfM procedures using a single-node in the ALC and OCC with different environments for (a) small and (b) large datasets. Lower is better.

### 3.3. Performance of Parallel Processing in Cluster Systems

Multi-nodes processing was tested by increasing the number of nodes from a single-node. As the number of GPU was limited up to six in the OCC, the shape with a single GPU, VM.GPU2.1 and VM.GPU3.1, were selected, and the performance of multi-node processing was compared with the ALC. SfM processing was conducted by Metashape in



exactly the same way as single-node processing, but Align Photos, Build Dense Cloud, Build DEM, and Build Orthomosaic were considered in comparison because Export DEM/Orthomosaic were still processed in a single-node. Figure 4 shows the absolute processing time with different cluster environments. As the nodes of the ALC were connected internally through the router in the isolated network, the ALC performed faster than the clusters of OCC. The network speed of the OCC could affect the disk I/O and communication between nodes for parallel processing. Nevertheless, the processing time with multiple nodes in the OCC decreased more rapidly when another node was added. The decreasing slope of both cluster systems converged with 5 nodes. As mentioned in Section 3.2, the multi-nodes with VM.GPU2.1 performed faster than VM.GPU3.1 for a large dataset. Although OCC took a longer processing time by more than two times in multi-nodes processing due to network speed, the results showed that cloud-based clusters could process UAS data using SfM software more efficiently.



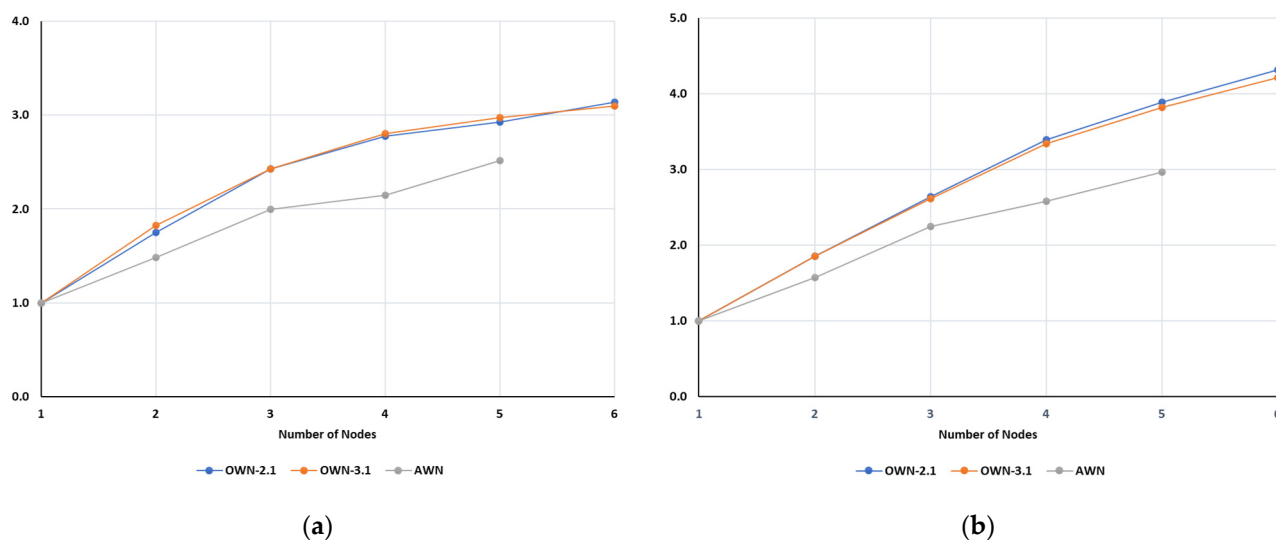
**Figure 4.** Processing time with the number of multi-nodes in the ALC and OCI for (a) small and (b) large datasets.

To compare how efficient the ALC and OCC were in multi-nodes processing, speed-up and efficiency were calculated using the processing time (Figures 5 and 6). Speed-up is defined as the ratio of the time taken to process data on a single-node to the time required to perform the same work on multiple nodes. In an ideal case, parallel processing could have a liner speed-up, 1-to-1 line, which means that the speed of execution increases with the number of nodes. Generally, the real speed-up is lower than the number of nodes, which means the slope should be lower than 1, and closer to 1-to-1 line is better. In this study, cloud-based clusters showed approximately 15–25 percent better performance of the speed-up algorithm in SfM processing for both small and large datasets. Since the nodes in the ALC were not uniform, speed-up was fluctuated more, while the speed-up values of the clusters in the OCC increased gradually. Regardless of the datasets, the clusters with VM.GPU2.1 and 3.1 had almost the same speed-up value (Figure 5).

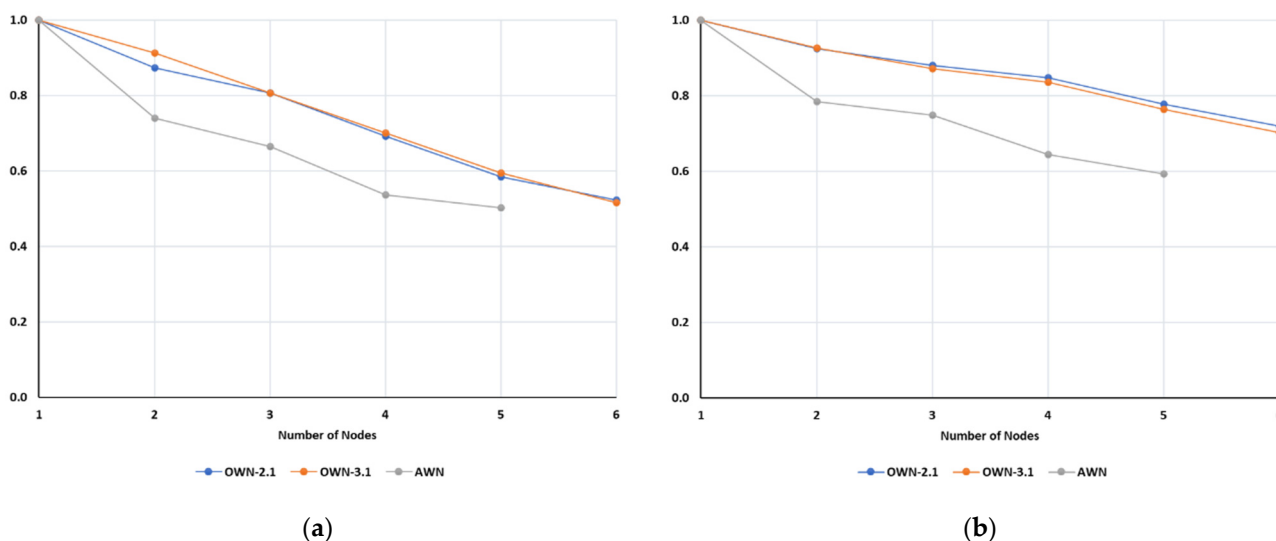
Efficiency is a performance metric estimating how well-utilized the nodes are processing data, compared to how much effort is wasted in communication and synchronization. Some nodes and the time in tasks can usually be wasted in either idling or communicating. Therefore, efficiency is lower than 1 in a real case and decreases with more nodes. Figure 6 shows the efficiency with the number of multi-nodes for small and large datasets. Similar to the speed-up results, the clusters in OCC showed better performance than the ALC and more stable efficiency with additional nodes. In particular, higher speed-up and efficiency were measured in the multi-nodes processing for the large dataset. These results imply that the cloud-based cluster could provide a better and more stable system for SfM processing when using the UAS data. If users would adopt the



appropriate number of nodes and shapes in the OCC, they could construct a more efficient and stable cluster system than an on-premise cluster.



**Figure 5.** Speedup with the number of multi-nodes in the ALC and OCI for (a) small and (b) large datasets.



**Figure 6.** Efficiency with respect to the number of multi-nodes in the ALC and OCI for (a) small and (b) large datasets.

#### 4. Conclusions

In this study, cloud computing- and local-cluster systems with various options were tested to compare the performance of SfM processing using UAS images collected in agricultural fields. Two UAS datasets were collected over the agricultural fields and processed by SfM software, Agisoft Metashape, with different computing environments. The performance of local machines and clusters were compared with cloud computing systems. Although local machine and cluster processed UAS datasets faster because of the network speed and disk I/O, cloud-based clusters showed better speed-up and efficiency in parallel processing. The experiments demonstrated that cloud computing could provide more stable and efficient systems to process massive UAS images when the user adopts the proper number and specification of nodes. In addition, cloud computing can give us the flexibility to increase instances more efficiently without having to worry about maintaining security or increasing capability. In the future, we will apply the cloud computing and cluster systems to process a huge dataset for various applications such as forest fire, coasting monitoring, environmental change detection, etc. in real/semi-real time.

**Author Contributions:** Conceptualization, Anjin Chang and Jinha Jung; methodology, Anjin Chang and Jinha Jung; software, Rajib Ghosh and Bryan Barker; validation, Anjin Chang and Jose Landivar; formal analysis, Anjin Chang and Jose Landivar; investigation, Anjin Chang; resources, Rajib Ghosh, Bryan Barker and Anjin Chang; data curation, Anjin Chang and Jose Landivar; writing—original draft preparation, Anjin Chang, Bryan Barker and Jose Landivar; writing—review and editing, Jinha Jung, Juan Landivar, Bryan Barker and Rajib Ghosh; visualization, Anjin Chang and Jose Landivar; supervision, Jinha Jung.; project administration, Juan Landivar; funding acquisition, Anjin Chang, Jinha Jung and Juan Landivar. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Data sharing is not applicable to this article.

**Acknowledgments:** This work was supported by Texas A&M AgriLife Research and in part by Oracle Cloud credits and related resources provided by the Oracle for Research program.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Appendix A

Specifications of the CPU and GPU equipped in each node are shown in Tables A1 and A2. A shape is a template that determines the number of OCPUs, amount of memory, and other resources that are allocated to an instance in the OCI. In this study, GPU shapes for virtual machines were adopted. An OCPU is defined as the CPU capacity equivalent of one physical core of an Intel Xeon processor with hyper-threading enabled, or one physical core of an Oracle SPARC processor. The previous generation VM shape, VM.GPU2.1, is not currently available.

**Table A1.** Specifications of CPU of the nodes in the AgriLife and OCI.

Cluster	Processor	# of Cores (Thread)	Lithography (nm)	Base Frequency (GHz)	Max Turbo Frequency (GHz)	Cache (MB)
AgriLife	Intel(R) Core(TM) i7-4790K	4 (8)	22	4.0	4.4	8
	Intel(R) Core(TM) i7-8700K	6 (12)	14	3.7	4.7	12
	Intel(R) Xeon(R) E5-1650	6 (12)	32	3.2	3.8	12
	Intel(R) Xeon(R) E5-2680	8 (16)	32	2.7	3.5	20
OCI	Intel(R) Xeon(R) Platinum 8167 M	26 (52)	14	2.0	2.4	36

**Table A2.** Specifications of GPU of the nodes in the AgriLife and OCI.

Cluster	Graphic Card	CUDA Cores	Bus Support	Base Clocks (MHz)	Memory (GB)
AgriLife	GeForce GTX 980	2048	PCI Express 3.0	1064	4
	GeForce GTX 1050 Ti	768	PCI Express 3.0	1290	4
	GeForce GTX 1070 Ti	2432	PCI Express 3.0	1607	8
OCI	NVIDIA Tesla P100	3584	PCI Express 3.0	1189	16
	NVIDIA Tesla V100	2150	PCI Express 3.0	1246	16

**Table A3.** Specifications of compute shapes in the OCI.

Shape	OCPU	CPU Memory (GB)	GPU Memory (GB)	Max Network Bandwidth
-------	------	-----------------	-----------------	-----------------------

				(Gbps)
VM.GPU2.1	12	72	16	8
VM.GPU3.1	6	90	16	4
VM.GPU3.2	12	180	32	8
VM.GPU3.4	24	360	64	24.6

## References

- Chang, A.; Jung, J.; Yeom, J.; Maeda, M.M.; Landivar, J.A.; Enciso, J.M.; Avila, C.A.; Anciso, J.R. Unmanned Aircraft System-(UAS-) Based High-Throughput Phenotyping (HTP) for Tomato Yield Estimation. *J. Sens.* **2021**, *2021*, 8875606. <https://doi.org/10.1155/2021/8875606>
- Xie, C.; Yang, C. A review on plant high-throughput phenotyping traits using UAV-based sensors. *Comput. Electron. Agric.* **2020**, *178*, 105731. <https://doi.org/10.1016/j.compag.2020.105731>
- Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A Review on UAV-Based Applications for Precision Agriculture. *Information* **2019**, *10*, 349. <https://doi.org/10.3390/info10110349>
- Chang, A.; Jung, J.; Yeom, J.; Landivar, J. 3D Characterization of Sorghum Panicles Using a 3D Point Cloud Derived from UAV Imagery. *Remote Sens.* **2021**, *13*, 282. <https://doi.org/10.3390/rs13020282>
- Garcia Millan, V.E.; Rankine, C.; Sanchez-Azofeifa, G.A. Crop Loss Evaluation Using Digital Surface Models from Unmanned Aerial Vehicles Data. *Remote Sens.* **2020**, *12*, 981. <https://doi.org/10.3390/rs12060981>
- Yeom, J.; Jung, J.; Chang, A.; Ashapure, A.; Maeda, M.; Maeda, A.; Landivar, J. Comparison of Vegetation Indices Derived from UAV Data for Differentiation of Tillage Effects in Agriculture. *Remote Sens.* **2019**, *11*, 1548. <https://doi.org/10.3390/rs11131548>
- Ashapure, A.; Jung, J.; Yeom, J.; Chang, A.; Maeda, M.; Maeda, A.; Landivar, J. A novel framework to detect conventional tillage and no-tillage cropping system effect on cotton growth and development using multi-temporal UAS data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 49–64. <https://doi.org/10.1016/j.isprsjprs.2019.04.003>
- Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios I. A compilation of UAV applications for precision agriculture. *Comput. Netw.*, **2020**, *172*, 107148. <https://doi.org/10.1016/j.comnet.2020.107148>
- Zaragoza, I.M.; Caroti, G.; Piemonte, A.; Riedel, B.; Tengen, D.; Niemeier, W. Structure from motion (SfM) processing of UAV images and combination with terrestrial laser scanning, applied for a 3D-documentation in a hazardous situation. *Geomat. Nat. Hazards Risk* **2017**, *8*, 1492–1504. <https://doi.org/10.1080/19475705.2017.1345796>
- Jung, J.; Landivar, J.; Chang, A.; Maeda, M.M.; Miller, A.D.; Kulasekaran, S.; Gabriel, G. Uashub: Building a Modern Cloud-Based Data Portal for the Management of UAS Big Data. Proceeding of 2020 ASA-CSSA-SSSA International Annual Meeting, Virtual, 9–13 November 2020.
- Compute Shapes in Oracle Cloud Infrastructure Documentation. Available online: <https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm> (accessed on 5 October 2021).
- Overview of File Storage in Oracle Cloud Infrastructure Documentation. Available online: <https://docs.oracle.com/en-us/iaas/Content/File/Concepts/filestorageoverview.htm> (accessed on 5 October 2021).
- Overview of Block Volume in Oracle Cloud Infrastructure Documentation. Available online: <https://docs.oracle.com/en-us/iaas/Content/Block/Concepts/overview.htm> (accessed on 5 October 2021).
- He, F.; Zhou, T.; Xiong, W.; Hashemininasab, S.M.; Habib, A. Automated Aerial Triangulation for UAV-Based Mapping. *Remote Sens.* **2018**, *10*, 1952. <https://doi.org/10.3390/rs10121952>
- Benassi, F.; Dall'Asta, E.; Diotri, F.; Forlani, G.; Morra di Cella, U.; Roncella, R.; Santise, M. Testing Accuracy and Repeatability of UAV Blocks Oriented with GNSS-Supported Aerial Triangulation. *Remote Sens.* **2017**, *9*, 172. <https://doi.org/10.3390/rs9020172>
- Buzbee, B.L. The Efficiency of Parallel Processing. *LOS ALAMOS SCIENCE* **1983**, *9*, 71.
- Ananth, Y.G.; Anshul, G.; Vipin, K. Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures. *IEEE Parallel and Distributed Technology: System and Technology* **1993**, *1*, 12–21. <https://doi.org/10.1109/88.242438>