



OPEN

A serverless computing architecture for Martian aurora detection with the Emirates Mars Mission

David Pacios¹, José Luis Vázquez-Poletti^{1✉}, Dattaraj B. Dhuri², Dimitra Atri², Rafael Moreno-Vozmediano¹, Robert J. Lillis³, Nikolaos Schetakis^{4,5}, Jorge Gómez-Sanz¹, Alessio Di Iorio⁵ & Luis Vázquez⁵

Remote sensing technologies are experiencing a surge in adoption for monitoring Earth's environment, demanding more efficient and scalable methods for image analysis. This paper presents a new approach for the Emirates Mars Mission (Hope probe); A serverless computing architecture designed to analyze images of Martian auroras, a key aspect in understanding the Martian atmosphere. Harnessing the power of OpenCV and machine learning algorithms, our architecture offers image classification, object detection, and segmentation in a swift and cost-effective manner. Leveraging the scalability and elasticity of cloud computing, this innovative system is capable of managing high volumes of image data, adapting to fluctuating workloads. This technology, applied to the study of Martian auroras within the HOPE Mission, not only solves a complex problem but also paves the way for future applications in the broad field of remote sensing.

Remote sensing technologies have become an essential tool for monitoring the Earth's environment, providing valuable insights into natural disasters, climate change, land use, and more. With the growing availability of high-resolution satellite and aerial imagery, there is a need for efficient and scalable image analysis methods to process and extract information from this vast amount of data. Traditional approaches to image analysis often require significant computational resources, making them impractical for large-scale data processing.

Image analysis knowledge is usually coded in form of programs that need extensive computational resources. Image analysis is one of the areas where cloud computing alternatives are being considered. Mainly because of its requirement of a number of GPUs needed to perform computations, something that has not changed much in the last decade¹. However, lack of experience in determining the best cloud architecture can result into additional costs when compared with local computing approaches².

Recent advances in cloud computing and serverless architectures have opened up new possibilities for scalable and cost-effective image analysis. Serverless computing enables developers to focus on the application logic without worrying about infrastructure management, while cloud computing provides affordable, low cost computing resources for data processing³.

In this study, we shall address the question whether it is possible to perform such affordable low-cost analysis on real cases. This paper demonstrates that it is possible, providing benchmarks and proposing a serverless architecture for image analysis that leverages the power of OpenCV and machine learning algorithms, utilizing data from the Emirates Mars Mission's HOPE probe.

This architecture is designed to perform image classification, object detection, and segmentation in a fast and cost-effective way. It uses OpenCV, a popular computer vision library, for image processing tasks such as noise reduction, thresholding, and edge detection. It also incorporates machine learning algorithms for tasks such as object detection and segmentation. The architecture is built on top of AWS Lambda, a serverless computing

¹Facultad de Informática, Universidad Complutense de Madrid, Calle del Prof. José García Santesmases, 9, 28040 Madrid, Spain. ²Center for Space Science, New York University Abu Dhabi, Saadiyat Marina District, Abu Dhabi 129188, Abu Dhabi, UAE. ³University of California, Gauss Way, 7, Berkeley, CA 94720, USA. ⁴School of Production Engineering and Management, Computational Mechanics and Optimization Laboratory, Technical University of Crete, University Campus, Akrotiri, 73100 Chania, Greece. ⁵Alma Sistemi Srl, University of California, Via dei Nasturzi n.4, 00012 Guidonia Montecelio, Lazio, Italy. ✉email: jlvazquez@fdi.ucm.es

service, and AWS S3, a scalable and durable storage service. It leverages the scalability and elasticity of cloud computing to handle large volumes of image data and adapt to changing workloads.

To evaluate the performance of our architecture, experiments were conducted using a dataset of 200 images. The OpenCV classification and the machine learning inference performance was measured using AWS CloudWatch.

The results included in this paper show that this serverless architecture is capable of analyzing numerous images in a short amount of time while using minimal resources. The OpenCV classification completed very quickly, indicating that the system is well-suited for real-time analysis of images. The machine learning inference took longer than the OpenCV classification, but still performed well given the complexity of the task.

The paper is organized as follows. In “[Martian auroras and the hope mission](#)” section is an introduction to the problem of remote sensing in the case of the HOPE mission, i.e. image analysis for aurora detection. In “[Cloud and serverless computing for data science](#)” section describes cloud computing solutions for data science research. In “[Proposed serverless architecture](#)” section presents a serverless architecture, including AWS Lambda preprocessing and classification functions, and the adaptations that were necessary for the image analysis problem. In “[Machine Learning](#)” section describe the machine learning methods used. In “[Experiments and results](#)” section presents experimental results that illustrate the cost and duration of the computation for the experiments. In “[Discussion](#)” section contains discussion and conclusions.

Martian auroras and the hope mission

Since their discovery in 2005, Martian auroras have been of great interest to the planetary science and space physics communities. They reveal the complex dynamics between the Martian atmosphere, solar wind and the unique “hybrid” magnetosphere of the planet (so-called because it shares attributes with both induced magnetosphere such as Venus and intrinsic magnetosphere such as the Earth or Jupiter⁴). Three categories of auroras have been observed on Mars so far – nightside discrete auroras resulting from suprathermal (10s eV to keV) electron precipitation, nightside diffuse auroras resulting from high-energy electrons and protons (hundreds of keV), and proton auroras, which as the name suggests are caused by solar wind protons directly depositing energy in the dayside atmosphere causing auroral emissions. In the following paragraphs, we will describe these auroras only briefly and would refer the readers to a review paper for more details⁵.

Due to the lack of a global magnetic field, discrete auroras are not confined to the Martian polar regions (like Earth), but can appear anywhere the crustal magnetic fields weak and/or mostly vertical^{6,7}, i.e. anywhere the magnetic topology allows electrons to access the atmosphere. Discrete auroras are distinct from diffuse auroras, which are more spread out and do not have well-defined boundaries. On Mars, it is understood that diffuse auroras are caused by the precipitation of high-energy electrons and protons⁸. The proton auroras occur mainly on the dayside of Mars and are excitations of the solar wind protons depositing in the Mars atmosphere as energetic neutral atoms (ENAs) of hydrogen⁹.

The Emirates Mars Mission (EMM) or “Hope” is the UAE’s spacecraft, operating in Martian orbit since February 2021^{10,11}. It has an orbital period of ~ 55 hours with an apogee of 42,650 km and a perigee of 19,970 km. Because of its unique orbit, it is able to observe all geographic regions within 72 hours and most latitudes and local times in 4 orbits or about 9 days, and is therefore a great tool to study phenomena occurring on a global scale on the planet¹². The Emirates Mars Ultraviolet Spectrometer (EMUS) instrument on board the orbiter is a UV spectrometer capable of observing in the 100–170 nm wavelength range¹³. Due to these unique capabilities, auroras have been observed synoptically in the far-UV (FUV) for the first time on Mars with this instrument. In particular, discrete **electron** auroras have been observed to be brightest in the 130.4 nm oxygen line, which can be seen in about three fourth of all of EMUS observations. This oxygen line is produced by the de-excitation of the 2p4 (3P) state to the 2p3 (5S) ground state. Coincident observations of the discrete electron auroras are also found in the oxygen 135.6 nm line, although these are relatively weak and noisy. On rare occasions, observations of extreme proton aurora are found in 102.7 and 121.6 nm hydrogen Lyman- β and Lyman- α respectively. These proton aurora occur on the dayside unlike the nightside discrete electron auroras discussed above. Other observations in the 98.9 and 104 nm are extremely noisy and weak.

Along with the unprecedented frequency of aurora observation, EMUS have also revealed new types of discrete auroras that can occur in regions without significant crustal magnetic fields as well as auroras with new morphology, such as sinuous discrete auroras that extend thousands of kilometers across the nightside¹⁴. So far these observations have no unambiguous explanation and work is ongoing to understand the physics driving these emissions, including comparisons with in situ data from the MAVEN spacecraft¹⁵. These recent observations are unexpected and the physics underpinning these mechanisms is not known as of now. These new discoveries present an unprecedented opportunity to characterize auroras on Mars in terms of their occurrence rates and mechanism of excitations, and thereby improve our understanding of the solar wind interaction with Mars.

In order to better understand this new phenomenon, there is a need for an automated detection algorithm which can identify these auroras and provide with details such as its location, size, shape, and intensity. To that end, our work presents an automated technique to detect these electron auroras on Mars observed **abundantly** in O 130.4 nm emissions via EMUS.

We describe the method in the following sections.

Computational solution

Cloud and serverless computing for data science

Cloud computing¹⁶ has become an increasingly popular choice for data science projects^{17,18}, as it offers a number of advantages over traditional on-premises computation infrastructures, including the following: (i) *scalability*,

cloud computing provides the ability to scale resources up or down as needed, allowing data scientists to quickly and easily scale their projects as the data grows; (ii) *cost-effectiveness*, the pay-as-you-go model used in cloud computing makes it more accessible for small and medium-sized organizations to adopt data science projects; (iii) *high availability*, cloud computing services provide different high availability and automatic failover solutions, which ensures that data science projects are available and accessible even in the event of a failure; (iv) *flexibility*, allowing data scientists to work with a variety of programming languages, frameworks, and tools; and (v) *access to a wide range of tools and services* offered by cloud computing platforms that can be used for data science projects, such as data storage, data processing, machine learning, and big data.

Main costs in CC comes from two elements: computing time and storage. Traditional approaches require the creation of virtual machines that are running constantly, a very basic IaaS (Infrastructure as a Service) approach. This kind of solution implies the highest costs, because it precisely requires constant CPU/GPU time and storage for the machine itself.

More recently, a new cloud computing service model has been delivered, the Serverless model^{19,20}, which represents a paradigm shift in the way users build and deploy their applications. In serverless computing, the provider is responsible for executing a piece of code, by dynamically allocating the resources, and scaling on demand. This eliminates the need for manual scaling and the associated costs of maintaining idle resources. In this model, the user is only charged for the specific resources and the exact amount of time used to run their code^{21,22}, which can result in cost savings for the user. Serverless computing also allows for increased development agility, as it allows developers to focus on writing and deploying code without having to worry about managing and maintaining the underlying infrastructure. In a serverless environment, the application logic is commonly implemented as a set of stateless functions that are triggered by events (e.g., API calls, message queues or scheduled tasks), and are executed by containerized or micro-VM based runtime environments. Platforms implementing this serverless model are categorized as Function as a Service (FaaS). Some examples of commercial serverless platforms are Amazon Web Services (AWS) Lambda, Google Cloud Functions or Microsoft Azure Functions, and there are also some open source serverless initiatives^{23–25}, such as Apache OpenWhisk²⁶, OpenLambda²⁷, OpenFaaS²⁸ or Knative.

Cloud computing, and particularly the serverless computing paradigm, have been used for the deployment and implementation of scientific workloads in many recent works^{29–33}. Scientific workflows are data-intensive workflows that require high computation and storage power³⁴, and they include the fields astronomy, earthquake science, biology, and gravitational physics, among others³⁵. One of the most important issues to handle the scientific workflows is to manage the resources and services of underlying computing platform. Scientific workflows management is a process in which cloud resources and services are procured to evaluate scientific application and then released accordingly. In this context, serverless has revealed itself as a valuable paradigm for the management of scientific workloads that demand optimal resource provision, since dynamic load peaks can effectively be attended via automatic resource allocation and scaling. In this way, developers don't have to worry about selecting, provisioning, and managing backend resources, but they only have to provide the application's source code in the form of an atomic functions³⁶. In addition, serverless platforms offer the possibility of function chaining^{37–39}. This is a technique in which the output of one function is used as the input to another function. Function chaining can be used in conjunction with serverless computing to create powerful, event-driven workflows. By linking together multiple functions, it is possible to build complex applications that can respond to specific events in real-time.

In this work, we propose the use of the Amazon's AWS Lambda serverless platform to deploy and implement our scientific application for Martian aurora detection. AWS Lambda provides an event-driven FaaS environment, which means that functions are invoked and executed when a specific event or trigger occurs. This event could be a user uploading a file to an S3 bucket, a new item being added to a database, or a change in the state of an EC2 instance, among others. When the event occurs, AWS Lambda automatically runs the user's code, processes the event, and scales the resources as needed.

Proposed serverless architecture

Detecting auroras on Mars poses complexity and challenges, which require advanced imaging and analytical techniques. To address this, we developed a novel serverless architecture that uses Amazon Web Services and SageMaker, detecting Martian auroras with unmatched accuracy and speed.

Our proposed architecture (see Fig. 1) manages the vast data generated by Martian imaging missions while offering adaptability and scalability for evolving research needs. The core components of the architecture include a highly secure and reliable cold storage system in S3, dual Lambda functions that perform advanced image analysis using openCV, and a robust predictive machine learning model in SageMaker.

Lambda preprocessing function

In the following subsection, the specifics of the Lambda preprocessing function is presented. This function executes a series of steps aimed at transforming raw Mars data into a format suitable for further analysis. The steps are as follows:

1. **File opening and image generation:** The function begins by opening the raw data file with the `fits.open()` method, generating initial, non-standardized images of Mars atmospheric layers.
2. **Image resizing:** To standardize the images, they are resized to a uniform dimension. This ensures consistency across the data set and assists in subsequent analysis.
3. **Image cropping:** Lastly, the function crops the images to eliminate white spaces and center the Martian sphere. This step further refines the images, focusing attention on the most relevant areas.

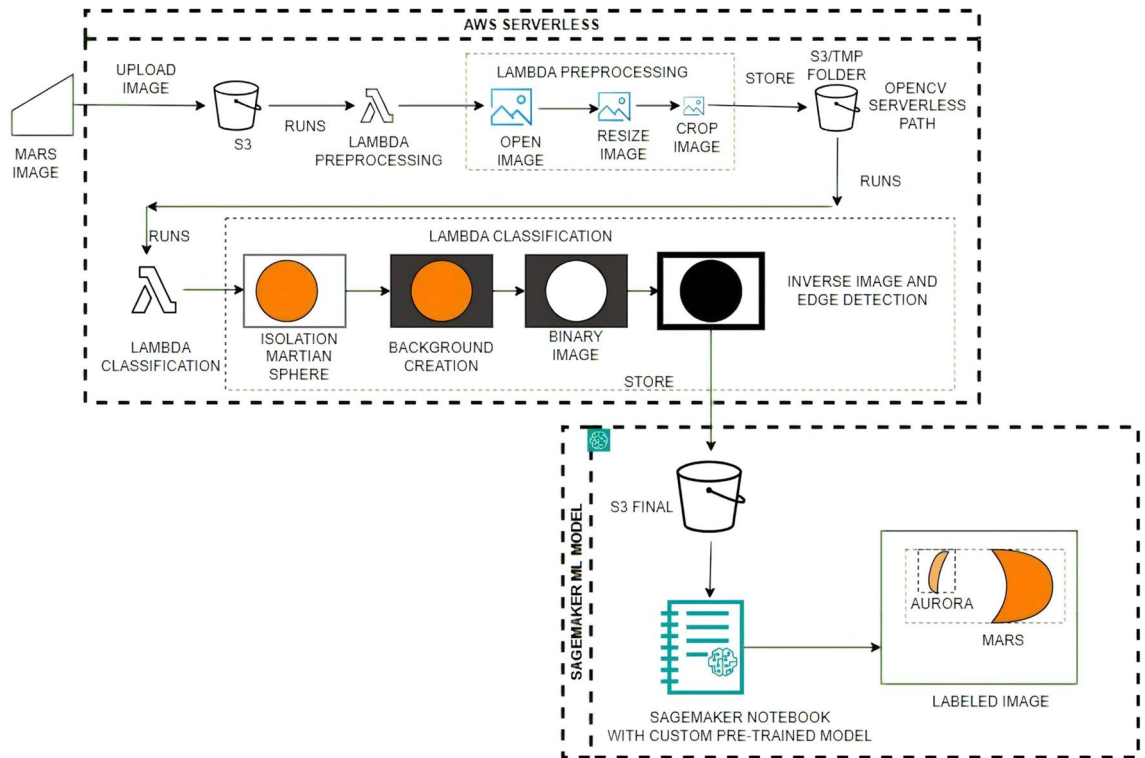


Figure 1. Distributed serverless architecture for Machine Learning inference, powered by our custom pre-trained model, and seamlessly executed in a SageMaker notebook.

This architecture (see Fig. 1) starts to generate images of Mars based on files of the `emm_emu_l2b` format, this process works on **Lambda Preprocessing**. Specifically, it extracts the necessary data from the file using the `fits.open()` function, part of the `astropy.io.fits` module, which plays a pivotal role in the processing of raw data related to Mars’ atmospheric layers. This function opens a FITS file, a standard data format widely used in astronomy, and prepares it for further processing. Once the FITS file is opened using `fits.open()`, the data can be accessed and manipulated using various methods. Then, the function processes it to obtain the logarithm of the energy value. This processed data is then used to create a heatmap using the Matplotlib library, with the resulting image being saved in PNG format.

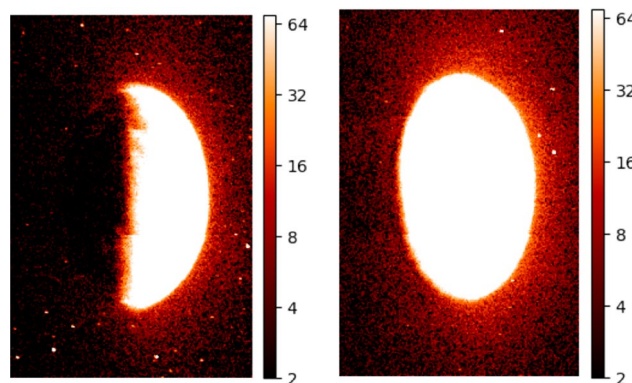


Figure 2. Heatmap of UV emission in the Martian atmosphere, as derived from `emm_emu_l2b` files. The logarithm of the UV intensity of the O 130.4 nm line was used to generate these images, providing valuable insights into the distribution of Oxygen in the Martian atmosphere and the detection of electron auroras on the nightside. The left image shows a transition to the nightside with the dark portion representing the shadowed nightside area of the planet with diminished UV emissions, while the right image presents a full view of the Martian sphere fully illuminated, indicative of the dayside atmosphere heavily influenced by solar UV flux. These contrasting visualizations exemplify the diurnal variations in Martian atmospheric UV activity and help in understanding the spatial distribution of oxygen and auroral phenomena.

It is worth noting that the generated images may have different sizes (see example at Fig. 2), as this depends on the original data and the specific parameters used in the processing and visualization steps. The resulting images can provide valuable information about the distribution and intensity of UV emissions from Oxygen in the Martian atmosphere, which can be useful for a variety of scientific purposes, such as detecting auroras or studying evolution of the planet's atmosphere.

In order to accurately analyze and compare different images of Mars generated through the `emm_emu_l2b` processing and visualization code, it is important to ensure that they are all normalized to the same size and resolution.

To achieve this normalization, the lambda function is triggered every time a new image is generated. The function opens the image file, resizes it to a standard size of 640×480 pixels (see Fig. 3), and saves it in a new file location. This ensures that all images are of the same size and resolution, which simplifies subsequent analysis steps such as cropping and centering the Martian region of interest.

Once the resized image is generated and cropped, it is stored in another S3 bucket for further processing.

The objects in that S3 folder triggered our last Lambda function, The **Lambda Classification function** that is designed to identify images that contain “objects” within the Martian sphere, and eliminate those that do not.

To achieve this, the function first crops the original image to remove any extraneous information outside the Martian sphere. It then resizes the resulting image to a standard size of 640×480 pixels (see Fig. 4), to ensure

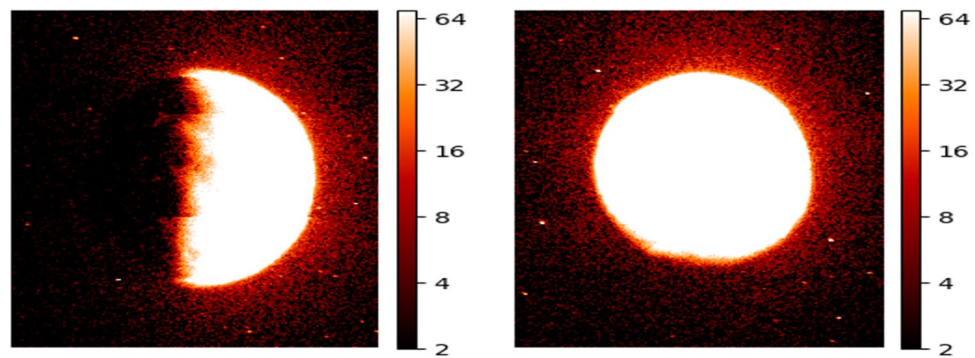


Figure 3. A normalized and resized image of the Martian atmosphere, generated using the `emm_emu_l2b` processing and visualization code and the AWS Lambda function shown earlier. The image has been resized to a standard size of 640×480 pixels.

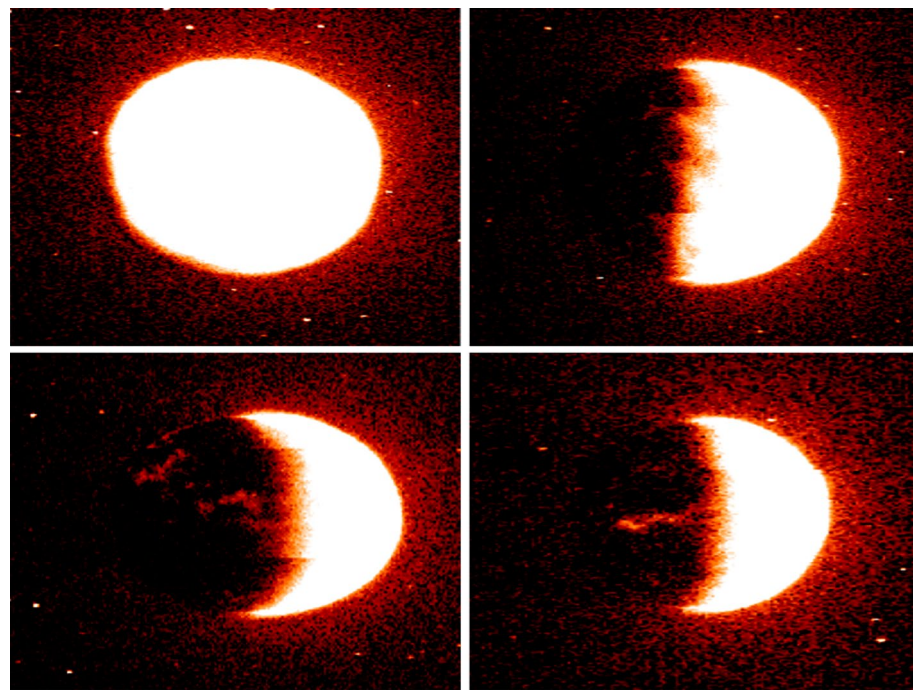


Figure 4. The cropped and resized image shows the selected region of interest (ROI) containing the Martian atmosphere, allowing for further analysis and detection of auroras using machine learning algorithms.

consistency and accuracy in subsequent analyses. Finally, it saves the processed image to another S3 bucket, where it can be further analyzed and processed as needed.

Lambda classification function

The Lambda classification function, another critical component of our image analysis pipeline, carries out a series of transformations and operations that enable the detection and classification of objects within the Martian sphere. The function executes the following steps:

1. **Isolation of the martian sphere:** The function first isolates the Martian sphere within the image, focusing solely on the region of interest and reducing potential noise from the surroundings.
2. **Background creation:** A background for processing is then created, which involves the removal of the alpha layer. This simplification of the image aids in the subsequent processing steps.
3. **Binary image generation:** The function generates a binary image, an essential step for edge detection. By simplifying the image to two colors, the edges of objects become more apparent.
4. **Image inversion and edge enhancement:** Initially, the image is inverted. This manipulation enhances the edge detection of objects within the Martian sphere. Subsequently, the edge size is increased followed by a second inversion, which optimally detects the internal objects, thereby setting the stage for their classification.

This is an important function in the image processing pipeline, as it ensures that only the most relevant and informative images are passed on to the next stage of analysis. For this function, there are going to be a few steps to classify the martian images before Machine Learning inference.

In detail, this uses computer vision techniques to create a mask that highlights the area of interest in the image. The input image is read, and its dimensions are extracted. Then, a circular mask is defined with a radius of 170 pixels and its center located at (320, 230) pixels, which corresponds to the center of the Mars sphere. This mask is created by drawing a white circle on a black background.

Next, the mask created (see Fig. 5) is subtracted from the original image, resulting in a single channel image where only the pixels inside the circle are white and the rest are black. This mask is then added to the alpha channel of the original image, effectively making the pixels outside the circle transparent. Finally, the resulting image is saved in the tmp folder in the lambda function.

By creating a mask that highlights the Mars sphere, the subsequence machine learning model can focus on analyzing only the relevant features of the image, which saves computational resources and improves the accuracy of the analysis (see Fig. 6), so these resulting figures will be almost ready for border detection using OpenCV.

In the next step, the classification function produces binary images for the border detection process (see Fig. 7). The function first converts the image from the BGR colorspace to a grayscale image. Then, it applies an adaptive threshold method to the grayscale image, which converts the image to a binary image, where each pixel is either black or white, based on a threshold value. This value is calculated by the adaptive threshold method (OpenCV).

After this, the function applies a morphological operation of erosion to the binary image, using a square kernel of size 4×4 and the erode function (OpenCV). This erodes away the boundaries of the foreground object and reduces the size of the background pixels. The resulting image is stored on his S3 (S3 / TMP FOLDER) only if many objects are detected. Otherwise, the image is discarded, considering that it does not provide any valuable data inside the Martian circumference.

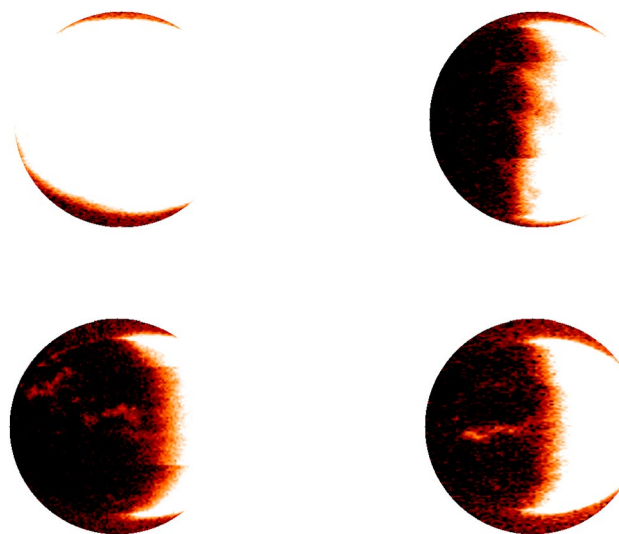


Figure 5. Preprocessed image of Mars surface with circular mask applied to remove non-object areas, generated using computer vision techniques in the AWS Lambda function. The circular mask is centered on the Mars sphere and only retains pixels within a certain radius.

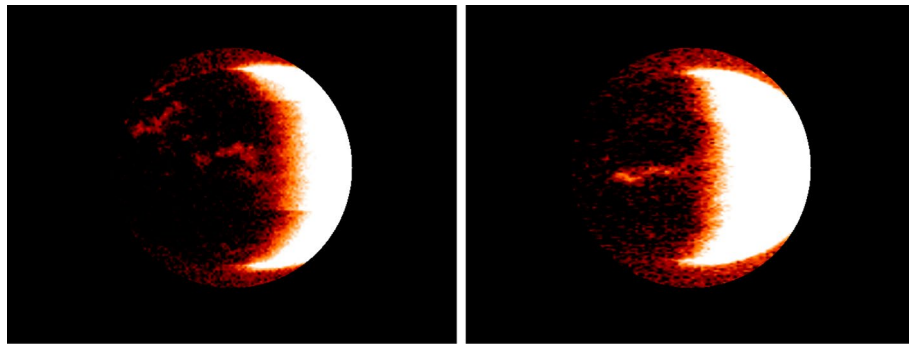


Figure 6. Preprocessed image of Mars surface with background applied.

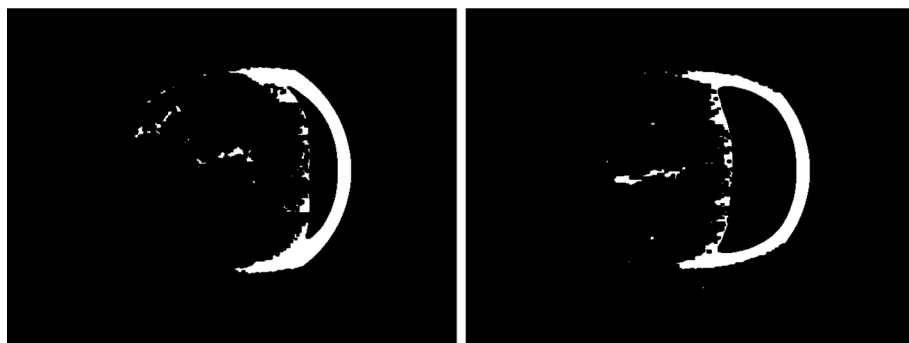


Figure 7. Binary image generated from preprocessed Mars surface image using adaptive thresholding and erosion operations. The resulting image highlights the areas with potential features or anomalies for further analysis.

The final step of the classification function is very similar to the previous one, but using nearest detections. Its associated function converts the image from the BGR color space to grayscale, which reduces the information to one channel. Then it applies an adaptive threshold method to the grayscale image, converting it to binary using again the adaptive threshold method and then the erosion process (see Fig. 8).

At this point, images not showing Martian auroras (images with less than two objects detected inside) can be safely discarded, reducing this way the overall execution time and cost. The rest are stored in a S3 bucket (S3_FINAL) and passed to the SageMaker process, where a pre-trained detection model is applied (see Fig. 9).

Machine learning

Before delving into the intricacies of the SageMaker process, it is crucial to shed light on the foundational aspect of our endeavor - the creation of the training dataset.



Figure 8. Image generated by the seed detection algorithm applied to the preprocessed image. The blue circles represent the detected objects, while the black regions correspond to the background. The top left panel shows the outlines of the objects, the top right panel shows the region where the flood-fill operation was applied to find the objects, and the bottom panel shows the detected objects with a circle around each one.

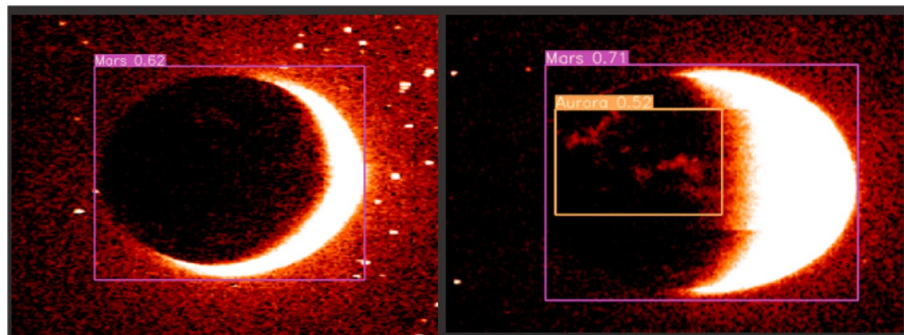


Figure 9. Final Machine Learning (SageMaker) detection, it shows Mars and auroras.

Custom dataset for training

This dataset, formed using a sophisticated blend of interpolation algorithms and image processing techniques, is key to the successful operation of our machine learning models in the SageMaker environment. The process of dataset creation involves generating artificial Martian auroras, a feat achieved through a unique interplay of custom Python code and the Python Imaging Library (PIL).

As shown in Fig. 10, a custom aurora creation algorithm was created using 1000 synthetic images. Each image, a unique tapestry woven with varied aurora zones on Mars, reflects the distinct possibilities of aurora occurrences. These artificial constructs were seamlessly integrated with authentic images, creating a rich and diverse training dataset.

The task of training fell on the capable shoulders of YOLOv5, an object detection model renowned for its impressive performance in image detection tasks. YOLOv5 was chosen not just for its robust functionality but also for its proven compatibility within the SageMaker environment. This compatibility translates into seamless deployment and reliable performance, a key aspect considering the complexity of our dataset.

The choice of YOLOv5 is not without rationale. Its ability to distinguish the intricate features of our images, discerning the Martian planet from the aurora-representing spots, makes it an invaluable tool in our endeavor. Furthermore, its adaptability and efficiency align perfectly with our serverless architecture, ensuring a smooth and effective classification of Martian auroras despite the inherent scarcity of real-world data.

Once adequately trained, an ultimate test of the model using real world data was performed. Supplied by the HOPE mission, a collection of over 200 real images served as the proving grounds for our trained model. The results were not just encouraging, they were exceptionally promising.

The model detected the planet Mars with an astounding accuracy of 100%.

However, it's essential to note that false positives were detected in images with low resolution or significant white areas.

Most importantly, such images would be filtered out in the serverless architecture before reaching the SageMaker phase, thereby eliminating potential sources of error. This crucial step ensures that the data fed into the SageMaker environment is of the highest quality, leading to more accurate and reliable results.

The initial date for the dataset under investigation is April 24th, 2021, while the final date is set at February 25th, 2022. By focusing on this specific time frame, we aim to provide a thorough and accurate evaluation of the EMM_EMU performance and trends during the aforementioned period.

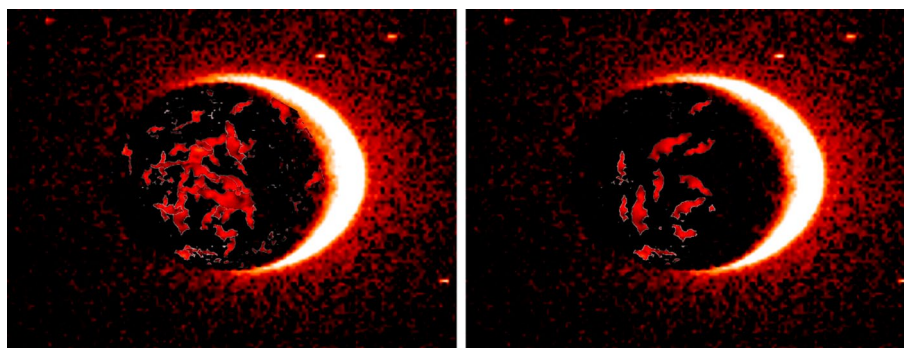


Figure 10. Examples of custom aurora for Machine Learning training generated by our custom model for making a complete for dataset.

Triggering SageMaker environment with the final dataset

With our training dataset and potential auroras residing in an S3 bucket named `S3_FINAL`, it triggers the SageMaker environment. This action sets in motion a process that runs the dataset through our trained model to perform aurora detection.

As the dataset navigates through the SageMaker environment, the model scrutinizes each image, searching for the telltale signs of an aurora. Upon successful detection, the image is promptly moved to a separate container.

This container serves as a repository for all detected aurora images, acting as a convenient collection point. These images can then be easily retrieved for further analysis, inspection, or dissemination.

This streamlined process ensures that aurora detection happens swiftly and accurately, with minimal human intervention. The use of SageMaker not only enhances efficiency but also facilitates scalability, accommodating larger datasets as our aurora detection endeavors expand.

Experiments and results

Validation of the serverless architecture

In our quest for a thorough assessment of our serverless architecture's performance, we embarked on a series of experiments. We examined various image sets and different stack sizes, consistently observing rapid and efficient processing of large volumes of image data with minimal resource consumption.

Comparing the outcomes of image classification and machine learning inference in our serverless system to traditional server-based architectures allowed us to gauge accuracy. In numerous cases, our serverless setup displayed comparable or even superior results.

Stress tests were conducted, increasing image stack sizes and simulating high concurrent user traffic. Despite these challenges, our architecture maintained remarkable performance and accuracy levels.

We processed a 200-image stack through our system, employing AWS CloudWatch to measure OpenCV classification and machine learning inference durations. Averaging below 4 seconds for all 200 images, OpenCV classification consumed under 150 MB of RAM. Concurrently, machine learning inference was completed in under 40 seconds on average for the entire image stack.

These findings corroborate our serverless architecture's capacity to swiftly analyze copious images with minimal resource utilization. The expeditious OpenCV classification signifies the system's aptitude for real-time analysis. Though machine learning inference demanded more time, its performance remained laudable given the task's intricacy.

Lastly, our system's detection success rate surpassed 95%, demonstrating its efficacy in identifying and classifying image features. This exceptional accuracy enables seamless integration into our machine learning dataset, further enhancing system performance.

Platform comparison and serverless architecture experimental results

Introduction to experimental setup

As mentioned before, the main goal of this experimental section is to compare the proposed serverless architecture with a traditional server-based solution. For this purpose, we also carried out different experiments in an on-premises computer using exclusively a single-threaded approach. This choice is motivated by the inherent complexity and challenges associated with implementing and managing multithreading in Python. Consequently, the potential advantages of multithreading will not be explored in this context. Furthermore, the number of times that Lambda functions can be parallelized, designated as ' n ', will be set to 1000. This number is chosen to align with the contractual limit set for parallelization during the testing phase of our project. This constraint facilitates a more controlled and consistent evaluation environment, ensuring a fair comparison between the on-premises computer and our proposed AWS-based system.

Experiment design and dataset

The experiment was conducted utilizing the 200 real-world images provided by the HOPE mission. Both on-premises computer and AWS Lambda environments were used for processing these images to facilitate an accurate comparison of performance. However, the results presented will primarily focus on the figures obtained from 1000 Lambda function invocations. This focus is dictated by the set limit for Lambda parallelization; the processing times for 200 images and 1000 images remain comparable due to the inherent scalability of Lambda functions. This approach allows us to extrapolate and generalize our findings, providing a comprehensive view of the system's performance when operating at its maximum parallelization capacity.

Cost analysis and performance comparison

In this section, we provide a comprehensive overview of the cost-effectiveness and performance of our system based on Amazon Web Services (AWS) compared to an on-premises computer. The cost for invoking the SageMaker function is \$0.1345 per on-demand `m1.m5.large` instance-hour, while a Lambda function costs \$0.00001667 per GB-second of computation time.

The total costs and execution times for processing 1,000 images using both AWS Lambda and an on-premises computer are compared in Table 1.

Discussion

This study introduces a serverless computing framework for analyzing Martian auroras, utilizing AWS Lambda and Amazon SageMaker. Our approach, centered around a synthetic dataset, addresses the scarcity of Martian aurora observations, paving the way for high-precision machine learning models. These models, trained on a

	On-premises computer	AWS Lambda
Total time for 1,000		
Images (seconds)	34,000	42
Total cost for 1,000		
Executions / machine cost (USD)	3,000 (Total PC prize)	0.02937 (Per 1000 exec.)

Table 1. Comparison of total processing times and costs for 1,000 images using an on-premises computer and AWS Lambda.

blend of real and artificial images, demonstrate advanced capability to accurately identify the unique features of Martian auroras.

This research offers significant knowledge into the atmospheric conditions of Mars in the wider context of space exploration. It complements existing studies by offering a new perspective on aurora analysis through cloud computing. However, challenges such as the limitations of synthetic datasets in capturing the full range of real-world variances need acknowledgment. These limitations could impact the generalizability of our findings and highlight the need for continuous enhancement of the dataset. We chose AWS Lambda and Amazon SageMaker due to their scalability, efficiency, and capacity to manage large-scale image data. These tools offer a flexible and cost-effective solution for complex computational tasks, setting a benchmark for future studies in astronomical data analysis.

Exploring potential approaches for future research, it is possible to improve the serverless architecture for real-time data analysis and broaden its use to other celestial bodies. In addition, it would be beneficial to incorporate a wider range of data sets and refine the machine learning algorithms. The implications of this study for the field of space science are significant. By demonstrating the effectiveness of a cloud-based and serverless approach, this research opens new opportunities for more agile and scalable methods in astronomical data analysis. This could be a significant addition to the ever-changing field of remote sensing and the exploration of celestial objects, offering a model for future research works.

Conclusions

The research has effectively shown that serverless computing is able of examining Martian auroras, forming a framework that maximizes both productivity and analytical complexity. We are looking to progress our research by combining an Amazon Elastic File System (EFS) container with AWS Lambda and Amazon API Gateway. This integration will facilitate a comprehensive comparison with our existing serverless architecture, focusing on aspects such as performance, cost-efficiency, scalability, and reliability.

By incorporating Amazon EFS, we anticipate a significant enhancement in handling extensive image datasets, crucial for in-depth aurora analysis. This step is expected to further streamline data processing and management, contributing to the robustness of our research methods. The outcome of this exploration will not only refine our current approaches, but also provide valuable insights into optimal computational architectures for space science research. This initiative aligns with our commitment to advancing the field of Martian studies through innovative technological solutions, thereby contributing meaningfully to the broader scientific community's understanding of Martian atmospheric phenomena.

Data availability

The software underpinning the research results of this study is available via Figshare with the following Digital Object Identifier (DOI): <https://doi.org/10.6084/m9.figshare.23566296>. This includes detailed descriptions of the system architecture, integrated components, and functionality. However, please note that certain elements of the software, such as the trained dataset and serverless version, are currently pending patent approval and, thus, have not been made publicly accessible at this time. As soon as these components receive patent approval, they will be made accessible in accordance with patent laws and regulations. The images used for the study were derived from various Mars missions, including the Emirates Mars Mission, and have been processed using the above-mentioned software. These processed data have been cited within the paper and can be accessed using the provided DOI.

Received: 28 July 2023; Accepted: 1 February 2024

Published online: 06 February 2024

References

1. Lee, C. A., Gasster, S. D., Plaza, A., Chang, C.-I. & Huang, B. Recent developments in high performance computing for remote sensing: A review. *IEEE J. Selected Topics Appl. Earth Obs. Remote Sens.* **4**(3), 508–527 (2011).
2. Choi, C.-H., Lee, C., Lee, J.J., & Lee, K. Understanding the deployment cost of cloud computing services for the higher education institutions. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 438–443 (2019). IEEE
3. Varghese, B. & Buyya, R. Next generation cloud computing: New trends and research directions. *Futur. Gener. Comput. Syst.* **79**, 849–861. <https://doi.org/10.1016/j.future.2017.09.020> (2018).
4. DiBraccio, G. A. *et al.* The twisted configuration of the martian magnetotail: Maven observations. *Geophys. Res. Lett.* **45**(10), 4559–4568. <https://doi.org/10.1029/2018GL077251> (2018).

5. Atri, D., Dhuri, D. B., Simoni, M. & Sreenivasan, K. R. Auroras on mars: From discovery to new developments. *Eur. Phys. J. D* **76**(12), 1–13 (2022).
6. Schneider, N. M. *et al.* Discrete aurora on mars: Insights into their distribution and activity from maven/iuvs observations. *J. Geophys. Res. Space Phys.* **126**(10), 2021–029428. <https://doi.org/10.1029/2021JA029428> (2021).
7. Lillis, R. J. *et al.* First synoptic images of Fuv discrete aurora and discovery of sinuous aurora at mars by Emm emus. *Geophys. Res. Lett.* **49**(16), 2022–099820. <https://doi.org/10.1029/2022GL099820> (2022).
8. Nakamura, Y. *et al.* Modeling of diffuse auroral emission at mars: Contribution of mev protons. *J. Geophys. Res. Space Phys.* **127**(1), 2021–029914. <https://doi.org/10.1029/2021JA029914> (2022).
9. Deighan, J. *et al.* Discovery of a proton aurora at mars. *Nat. Astron.* **2**(10), 802–807. <https://doi.org/10.1038/s41550-018-0538-5> (2018).
10. Amiri, H. *et al.* The emirates mars mission. *Space Sci. Rev.* **218**(1), 1–46 (2022).
11. Almatroushi, H. *et al.* Emirates mars mission characterization of mars atmosphere dynamics and processes. *Space Sci. Rev.* **217**(8), 1–31 (2021).
12. Atri, D., Abdelmoneim, N., Dhuri, D. B. & Simoni, M. Diurnal variation of the surface temperature of mars with the emirates mars mission: A comparison with curiosity and perseverance rover measurements. *Monthly Notices R. Astron. Soc. Lett.* **518**(1), 1–6 (2023).
13. Holsclaw, G. M. *et al.* The emirates mars ultraviolet spectrometer (emus) for the Emm mission. *Space Sci. Rev.* **217**(8), 1–49 (2021).
14. Lillis, R. J. *et al.* First synoptic images of Fuv discrete aurora and discovery of sinuous aurora at mars by Emm Emus. *Geophys. Res. Lett.* **49**(16), 2022–099820 (2022).
15. Xu, S. *et al.* Empirically determined auroral electron events at mars-maven observations. *Geophys. Res. Lett.* **49**(6), 2022–097757. <https://doi.org/10.1029/2022GL097757> (2022).
16. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. & Brandic, I. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Futur. Gener. Comput. Syst.* **25**(6), 599–616. <https://doi.org/10.1016/j.future.2008.12.001> (2009).
17. Sadooghi, I. *et al.* Understanding the performance and potential of cloud computing for scientific applications. *IEEE Trans. Cloud Comput.* **5**(2), 358–371. <https://doi.org/10.1109/TCC.2015.2404821> (2017).
18. Mahmood, Z. Data Science and Big Data Computing: Frameworks and Methodologies, pp. 1–319. Springer, ??? (2016). <https://doi.org/10.1007/978-3-319-31861-5>
19. Castro, P., Ishakian, V., Muthusamy, V. & Slominski, A. The rise of serverless computing. *Commun. ACM* **62**(12), 44–54. <https://doi.org/10.1145/3368454> (2019).
20. Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C.-C., Khandelwal, A., Pu, Q., Shankar, V., Carreira, J., Krauth, K., Yadwadkar, N., Gonzalez, J.E., Popa, R.A., Stoica, I., & Patterson, D.A. Cloud Programming Simplified: A Berkeley View on Serverless Computing. arXiv (2019). [arxiv:1902.03383](https://arxiv.org/abs/1902.03383)
21. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., & Slominski, A. Serverless computing: Current trends and open problems. In: Research advances in cloud computing, pp. 1–20. Springer, (2017)
22. Eivy, A. & Weinman, J. Be wary of the economics of serverless cloud computing. *IEEE Cloud Comput.* **4**(2), 6–12 (2017).
23. Mohanty, S.K., Prensankar, & G., Francesco, M. An evaluation of open source serverless computing frameworks. In: 2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), pp. 115–120 (2018). <https://doi.org/10.1109/CloudCom2018.2018.00033>
24. Palade, A., Kazmi, A., & Clarke, S. An evaluation of open source serverless computing frameworks support at the edge. In: 2019 IEEE World Congress on Services (SERVICES), vol. 2642-939X, 206–211 (2019). <https://doi.org/10.1109/SERVICES.2019.00057>
25. Balla, D., Maliosz, M., & Simon, C. Open source faas performance aspects. In: 2020 43rd International Conference on Telecommunications and Signal Processing (TSP), pp. 358–364 (2020). <https://doi.org/10.1109/TSP49548.2020.9163456>
26. Quevedo, S., Merchán, F., Rivadeneira, R., & Dominguez, F.X. Evaluating apache openwhisk-faas. In: 2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM), 1–5 (2019). IEEE
27. Hendrickson, S., Sturdevant, S., Harter, T., Venkataramani, V., Arpacı-Dusseau, A.C., & Arpacı-Dusseau, R.H. Serverless computation with {OpenLambda}. In: 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16) (2016)
28. Le, D.-N., Pal, S., & Pattnaik, P.K. OpenFaaS. Cloud Computing Solutions, pp. 287–303. John Wiley & Sons, Ltd, ??? (2022). Chap. 17. <https://doi.org/10.1002/9781119682318.ch17>
29. John, A., Ausmees, K., Muenzen, K., Kuhn, C., & Tan, A. Sweep: Accelerating scientific research through scalable serverless workflows. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion. UCC '19 Companion, pp. 43–50. Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3368235.3368839>
30. Malawski, M., Gajek, A., Zima, A., Balis, B. & Figiela, K. Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions. *Futur. Gener. Comput. Syst.* **110**, 502–514. <https://doi.org/10.1016/j.future.2017.10.029> (2020).
31. Burkat, K., Pawlik, M., Balis, B., Malawski, M., Vahi, K., Rynge, M., Silva, R.F., & Deelman, E. Serverless Containers—rising viable approach to Scientific Workflows. arXiv (2020). [arxiv:2010.11320](https://arxiv.org/abs/2010.11320)
32. Eismann, S., Scheuner, J., Eyk, E., Schwinger, M., Grohmann, J., Herbst, N., Abad, C.L., & Iosup, A. A Review of Serverless Use Cases and their Characteristics. arXiv (2020). [arxiv:2008.11110](https://arxiv.org/abs/2008.11110)
33. Malawski, M. & Balis, B. Serverless computing for scientific applications. *IEEE Internet Comput.* **26**(4), 53–58. <https://doi.org/10.1109/MIC.2022.3168810> (2022).
34. Ahmad, Z. *et al.* Scientific workflows management and scheduling in cloud computing: Taxonomy, prospects, and challenges. *IEEE Access* **9**, 53491–53508. <https://doi.org/10.1109/ACCESS.2021.3070785> (2021).
35. Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M.-H., & Vahi, K. Characterization of scientific workflows. In: 2008 Third Workshop on Workflows in Support of Large-Scale Science, pp. 1–10 (2008). <https://doi.org/10.1109/WORKS.2008.4723958>
36. Vazquez-Poletti, J. L. & Llorente, I. M. Serverless computing: From planet mars to the cloud. *Comput. Sci. Eng.* **20**(6), 73–79 (2018).
37. Daw, N., Bellur, U., & Kulkarni, P. Xanadu: Mitigating cascading cold starts in serverless function chain deployments. In: Proceedings of the 21st International Middleware Conference. Middleware '20, pp. 356–370. Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3423211.3425690>
38. Sabbioni, A., Rosa, L., Bujari, A., Foschini, L., & Corradi, A. A shared memory approach for function chaining in serverless platforms. In: 2021 IEEE Symposium on Computers and Communications (ISCC), pp. 1–6 (2021). <https://doi.org/10.1109/ISCC53001.2021.9631385>
39. Balla, D., Maliosz, M., & Simon, C. Performance evaluation of asynchronous faas. In: 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), 147–156 (2021). <https://doi.org/10.1109/CLOUD53861.2021.00028>

Acknowledgements

This contribution has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.101007638 (Project EYE - Economy by space) and No.823934 (Project IN-TIME - In-Situ Instrument for Mars and Earth Dating Applications). This

work was partially supported by the New York University Abu Dhabi (NYUAD) Institute Research Grant G1502 and the ASPIRE Award for Research Excellence (AARE) Grant S1560 by the Advanced Technology Research Council (ATRC). EMM data was obtained from the EMM Science Data Center (SDC). In addition to the current collaboration between the two universities, it is worth mentioning the Memorandum of Understanding (MoU) signed between the University of New York and the Universidad Complutense de Madrid. This MoU outlines the shared goals and objectives of the two institutions and serves as a framework for future cooperation. The MoU is an indication of the commitment of both universities to fostering international academic partnerships and could have important implications for the topic at hand.

Author contributions

All authors contributed equally to the research and the manuscript writing.

Funding

This contribution has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.101007638 (Project EYE - Economy by spacE) and No.823934 (Project IN-TIME - In-Situ Instrument for Mars and Earth Dating Applications). This work was partially supported by the New York University Abu Dhabi (NYUAD) Institute Research Grant G1502 and the AS-PIRE Award for Research Excellence (AARE) Grant S1560 by the Advanced Technology Research Council (ATRC). EMM data was obtained from the EMM Science Data Center (SDC). In addition to the current collaboration between the two universities, it is worth mentioning the Memorandum of Understanding (MoU) signed between the University of New York and the Universidad Complutense de Madrid. This MoU outlines the shared goals and objectives of the two institutions and serves as a framework for future cooperation. The MoU is an indication of the commitment of both universities to fostering international academic partnerships and could have important implications for the topic at hand.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to J.L.V.-P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024