SpotVerse: Optimizing Bioinformatics Workflows with Multi-Region Spot Instances in Galaxy and Beyond

Myungjun Son
The Pennsylvania State University
Pennsylvania, USA
mjson@psu.edu

Gulsum Gudukbay
The Pennsylvania State University
Pennsylvania, USA
gulsum@psu.edu

Mahmut Kandemir
The Pennsylvania State University
Pennsylvania, USA
mtk2@psu.edu

ABSTRACT

As demand for cloud computing in bioinformatics increases, various studies have explored options for running large-scale workloads with reduced costs, often leveraging spot instances in multi-region deployments. For example, spot instances offer lower prices but come with the risk of interruption, contrasting with regular (on-demand) instances. However, transitioning to regions with high interruption rates can undermine the benefits of spot instances, adversely affecting performance and cost efficiency. Additionally, regular instances sometimes outperform spot instances based on their specifications. Existing IaaS frameworks focus primarily on cost savings without adequately addressing performance stability in high-interruption regions. To address these challenges, we introduce SpotVerse, a framework designed to optimize cloud resource allocation for bioinformatics workloads, including those within Galaxy – an open-source, web-based platform widely used for managing bioinformatics workflows. SpotVerse efficiently manages long workloads at reduced costs while navigating the complexities of high-interruption regions and strategically selecting between on-demand and spot instances. Our experiments compare SpotVerse with traditional single-region deployments, on-demand instances, and other existing frameworks to evaluate its performance and cost efficiency. Through advanced algorithms for resilient workflows and heuristic resource management, SpotVerse minimizes disruption risks and showcases potential cost savings of up to 52% over traditional single-region deployments.

CCS CONCEPTS

• **Computer systems organization → Cloud computing**.

KEYWORDS

resource-management, spot instances, multi-region

ACM Reference Format:
Myungjun Son, Gulsum Gudukbay, and Mahmut Kandemir. 2024. SpotVerse: Optimizing Bioinformatics Workflows with Multi-Region Spot Instances in Galaxy and Beyond. In *24th International Middleware Conference (MIDDLEWARE '24), December 2–6, 2024, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3652892.3700750

MIDDLEWARE '24, December 2–6, 2024, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-0623-3/24/12
https://doi.org/10.1145/3652892.3700750

1 INTRODUCTION

Numerous entities are increasingly capitalizing on the capabilities of cloud computing, marking a shift in how data is processed, stored, and accessed. Cloud ecosystems have evolved to provide a diverse set of offerings that surpass traditional infrastructure models, including not only foundational services such as Infrastructure-as-a-Service (IaaS) [48], which offers compute resources and storage, but also extensions to comprehensive solutions such as Software-as-a-Service (SaaS) [36] for remote application access, and Function-as-a-Service (FaaS) [47] for executing code snippets in a serverless environment. Given the inherent flexibility, scalability, and accessibility these platforms offer, they have become a popular choice for running a wide range of applications.

Considering the demands of bioinformatics for extensive computational resources and long execution times [55, 56, 80, 83, 102], cloud computing offers a compelling solution. Its versatility, scalability, and convenience align well with the needs of bioinformatics research, facilitating widespread adoption in the field [56, 65, 75, 86, 98]. While our primary focus in this work is on bioinformatics workloads due to their characteristically long durations, the principles and techniques discussed herein are adaptable to other workloads as well, highlighting the flexibility and broader applicability of our proposed solutions.

Within this landscape, *Galaxy* [1] emerges as a leading framework in bioinformatics. Galaxy simplifies complex data analyses, making them accessible to researchers without deep computation expertise. As an open-source, web-based workflow management system, it finds extensive use in fields such as drug discovery, genome assembly, machine learning, and computational chemistry [51, 52, 57, 61, 62, 70, 71, 99, 101]. Its user-friendly interface and ease of use make it a crucial tool in bioinformatics, facilitating complex analytics and research. Galaxy can be deployed within major cloud environments such as AWS [9], Google Cloud [20], and Microsoft Azure [22], leveraging their scalability and computational power for efficient large-scale analyses. Unlike general-purpose cloud services, Galaxy offers a specialized framework with pre-configured tools and workflows tailored for bioinformatics, catering uniquely to the needs of researchers in this field.

As bioinformatics evolves, leveraging cloud computing for cost optimization and performance enhancement has become a critical concern, leading cloud providers to offer different types of instances to meet varying needs. "On-demand instances" provide guaranteed availability at a fixed rate, making them reliable but more expensive [25]. In contrast, "spot instances" are basically underutilized cloud resources offered at a reduced price but with the risk of potential interruption during execution [6]. Spot instances are dynamically priced and can be terminated by the cloud provider

when they need the capacity back, offering a cost-effective solution for non-urgent or fault-tolerant workloads [50].

Furthermore, beyond individual spot instances, the concept of multi-region deployments offers significant advantages over traditional single-region offerings by cloud providers. Recent research [53, 77, 84, 87, 104, 106] has demonstrated that such deployments are not only feasible but also cost-effective, distributing tasks across multiple geographical regions within a cloud provider's network. This strategy enhances service reliability and cost efficiency for bioinformatics tasks, particularly for parallel and long-running tasks, by mitigating the risks associated with service interruptions and regional outages while potentially reducing costs.

Despite the benefits, multi-region deployments using spot instances face significant challenges, particularly the risk of inadvertently directing tasks to regions prone to higher spot instance interruptions. Such occurrences can escalate latency and negatively impact the overall efficiency of workflows. Additionally, existing IaaS frameworks have limitations in managing spot instances, often focusing primarily on cost savings without adequately addressing performance stability in high-interruption regions. These issues underscore the need for refined strategies to manage multi-region deployments effectively. Moreover, specific scenarios might necessitate the selection of on-demand instances over spot instances, depending on specific organizational objectives. This calls for sophisticated algorithms capable of *intelligently* allocating tasks between spot and on-demand instances, achieving an "optimal mix" of cost-efficiency and reliability for diverse needs in multi-region cloud deployments.

To address the aforementioned challenges, we propose **SpotVerse**, a multi-region cloud software specifically designed to manage bioinformatics workloads, including those hosted on Galaxy. SpotVerse leverages novel algorithms and heuristics to improve cost-efficiency and reliability over existing cloud management solutions. Our study in this work has two main goals. First, we create a system naturally resistant to failures, designed for general bioinformatics workloads and specific bioinformatics tasks within Galaxy frameworks in a multi-region setting. Second, we develop a practical algorithm to efficiently distribute cloud resources for the required tasks, offering advantages over current approaches. The **key contributions** of this paper are thus as follows:

- We introduce SpotVerse, demonstrating its proficiency in managing bioinformatics workflows within a multi-region framework, particularly adept at handling the unpredictability of spot instance interruptions.
- We integrate a heuristic algorithm into SpotVerse for optimal resource allocation across multiple regions, prioritizing factors such as "Interruption Frequency" and "Spot Placement Score" beyond just spot prices.
- We ensure SpotVerse's resilience by demonstrating its applicability across various cloud services, allowing consistent performance despite spot instance variability.
- We demonstrate SpotVerse's adaptability and efficiency in multi-region settings through comprehensive experiments across various regions, instance sizes, instance types, and diverse bioinformatics workloads. Our empirical analysis, including comparisons against traditional single-region and on-demand instances, as well as state-of-the-art frameworks for spot instance management, reveals that SpotVerse can achieve cost reductions of up to 52% compared to the traditional single-region deployments.

This paper is structured as follows: Section 2 introduces the fundamental concepts behind SpotVerse and showcases its significance through motivating experiments. Section 3 delves into SpotVerse's design, while Section 4 outlines its practical implementation. Section 5 evaluates SpotVerse's performance and compares it with existing frameworks. Section 6 explores relevant prior research in the field. Section 7 discusses potential directions for future research and enhancements to the SpotVerse framework. Section 8 concludes the paper by summarizing our significant findings.

2 BACKGROUND AND MOTIVATION

2.1 Background

We begin our discussion by focusing on bioinformatics and the Galaxy framework designed to handle bioinformatics workflows. Next, we explore the use of spot instances and multi-region strategies. Following that, we delve into a motivational experiment and analysis, paving the way for SpotVerse, the solution we developed to fully harness these technologies while addressing their inherent challenges.

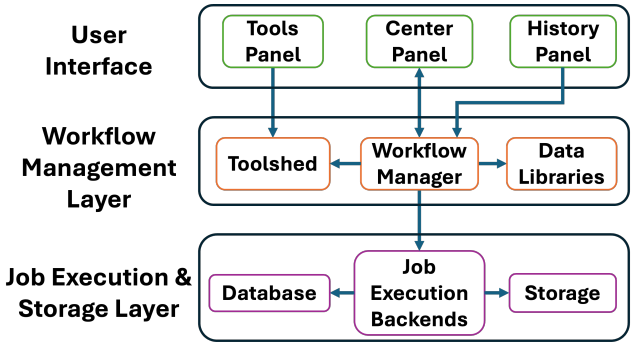
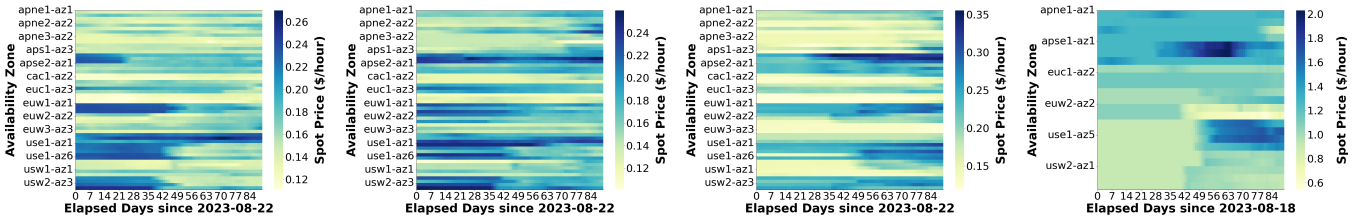


Figure 1: Schematic diagram of Galaxy's architecture. The User Interface layer allows interaction through the Tools Panel, Center Panel, and History Panel, while the Workflow Management layer orchestrates tasks by managing tools and datasets from the Toolshed and Data Libraries.

2.1.1 Bioinformatics Workloads in the Cloud. Bioinformatics focuses on processing and analyzing vast genomic datasets to decode biological information, such as DNA sequences, driving progress in medical, agricultural, and environmental research [68, 81, 94, 97, 100]. These processes necessitate robust computational frameworks capable of managing prolonged tasks. Numerous frameworks support bioinformatics workloads, including QIIME 2 for microbial ecology, DeepVariant (developed by Google) for variant calling, and TensorFlow for bioinformatics applications [15, 44, 59].

Among the various frameworks available, Galaxy enables sophisticated bioinformatics investigations within cloud environments. Developed to simplify complex data analyses for researchers without deep computing expertise, Galaxy is a widely used, open-source, web-based workflow management framework supporting diverse



(a) c5.2xlarge (compute-optimized) (b) m5.2xlarge (general-purpose) (c) r5.2xlarge (memory-optimized) (d) p3.2xlarge (GPU-optimized)

Figure 2: Spot price diversity across a spectrum of instance types and regions.

research fields. As shown in Figure 1, Galaxy's architecture is centered around key components such as the user interface, workflow manager, toolshed, and data libraries, which collectively provide a flexible and scalable framework for managing and executing diverse bioinformatics workflows.

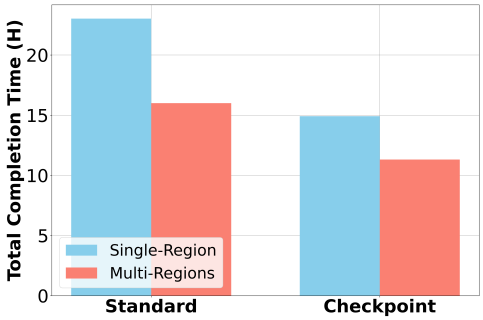
Galaxy can be deployed as a "container" using technologies like Docker [90] or Singularity [82] and can integrate with Kubernetes [63] environments for scalable and flexible deployment. This flexibility ensures that Galaxy can handle a range of computationally demanding tasks, from DNA sequencing to complex ML applications, particularly in genomics and proteomics, within cloud environments [57, 61, 62, 70, 71, 73, 99, 101].

While Galaxy excels in supporting bioinformatics workflows through its streamlined interface, it encounters challenges in optimizing cloud resource allocation and controlling costs, particularly when utilizing spot instances across multiple regions. SpotVerse addresses these limitations by optimizing cloud resource usage, specifically through efficient management of spot instances to reduce costs while ensuring performance and reliability. In addition to optimizing bioinformatics workflows within the Galaxy framework, SpotVerse extends its capabilities beyond Galaxy, managing standard bioinformatics tasks and various workloads.

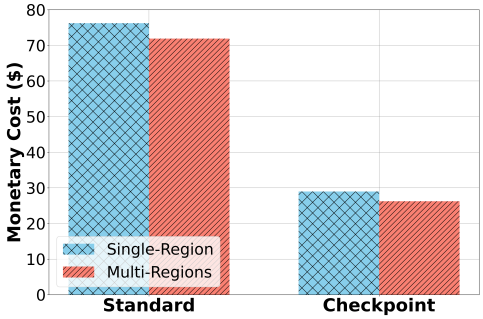
2.1.2 Spot Instances and Multi-Region Execution. For this study, understanding the details of cloud computing, especially the role of "spot instances", is essential. Spot instances, offered by cloud services such as Microsoft [14], AWS [6], and Google [72], have transformed how computing resources are accessed in the cloud. Unlike on-demand instances, which offer guaranteed availability at a fixed price, spot instances introduce a more dynamic model, providing surplus computing power at reduced costs by allocating unused capacity, thereby enhancing cost-effectiveness.

In the context of spot instances, "interruption" denotes the termination of these instances by the cloud provider due to a range of factors [39]. The primary reasons for spot instance interruptions include capacity requirements, where the cloud provider reclaims instances for reallocating capacity or operational needs such as host maintenance or hardware decommissioning. Interruptions can also occur due to pricing factors, particularly if the spot market price rises above the user's predetermined limit. Users receive a two-minute warning [38] before an interruption, offering a short period to manage disruptions. Although using spot instances requires strategies to manage their unpredictable nature, they significantly benefit from reduced costs.

In parallel, there is a growing trend in cloud computing towards "multi-region" and "multi-cloud" approaches. Many organizations recognize the advantages of distributing workloads across multiple regions to enhance redundancy, availability, and cost optimization [53, 77, 87, 104, 106]. While our analysis recognizes the growing importance of multi-cloud strategies involving multiple cloud providers, this paper specifically concentrates on the multi-region aspect within a single cloud provider. We have focused on collecting and analyzing instance price data from various regions to provide deeper insights into effective cloud resource management.



(a) Completion time



(b) Monetary cost

Figure 3: Workload completion time and cost: single vs. multi-region deployment.

In our analysis of cloud resource management, we evaluated representative instances: *c5.2xlarge* (compute-optimized), *r5.2xlarge* (memory-optimized), *m5.2xlarge* (general-purpose), and *p3.2xlarge*

(accelerated computing). As depicted in Figure 2, we observed significant fluctuations in spot instance prices across different AWS regions and Availability Zones (AZs) [30]. Each AZ is a distinct location within a data center region, designed to operate independently for high availability and reliability. The figure plots region and AZ ID combinations on the y-axis, elapsed days on the x-axis, and spot prices on the secondary y-axis.

These variations underscore the complex interplay between demand and supply in cloud services, emphasizing the importance of a multi-region distribution strategy for cost-effective cloud operations. Such an approach, necessitated by observed price volatility, enables strategic workload shifting between regions to capitalize on favorable pricing and mitigate risks associated with regional disruptions. Leveraging the economic diversity of different regions can enhance system resilience and yield significant cost savings, as demonstrated by the regional spot instance price variations in our study. To further explore the practical implications of these strategies, our empirical study delves into specific bioinformatics workloads within the Galaxy framework, aiming to quantitatively assess the benefits and challenges of multi-region deployment.

2.2 Motivational Experiment

Our empirical study leveraged insights from spot price variations and the benefits of distributing workloads across multiple regions. We specifically examined Genome Reconstruction and Next Generation Sequencing (NGS) Data Preprocessing workloads within the Galaxy framework. To assess the impact of multi-region distribution, we executed 42 workloads across three distinct regions: *ap-northeast-3*, *ca-central-1*, and *eu-north-1*, all utilizing *m5.xlarge* instances. We selected *ca-central-1* as our baseline for comparison due to its lowest cost for *m5.xlarge* instances across regions in single-region deployments. This setup enabled a comprehensive assessment of workloads under two operational categories:

• **Standard Workload (Genome Reconstruction):** Operates within a 10-hour window and requires complete re-execution from the start in case of interruptions.

• **Checkpoint Workload (NGS Data Preprocessing):** Operates within a 10-hour timeframe and resumes processing from the most recent checkpoint upon interruption.

In response to interruptions in any region, new spot instances were launched to either restart or resume the affected workloads, depending on their type. We closely monitored these diverse workload types, recording data on interruptions, total completion times, and associated costs.

Figure 3 presents a comparative analysis, *highlighting the benefits of multi-region over single-region strategies.* Specifically, for Standard Workloads, transitioning to a multi-region approach resulted in cost savings of about 5.67% and a substantial 30.49% decrease in total completion time. In contrast, Checkpoint Workloads saw a 9.43% cost reduction and a 6.63% reduction in completion time under the multi-region approach. Furthermore, the number of interruptions was significantly reduced: Standard Workloads experienced a 13.2% reduction (from 190 to 165), while Checkpoint Workloads saw a 41.6% decrease (from 125 to 73).

However, it is important to note that this experimental analysis does not capture all possible scenarios. There are instances where a multi-region strategy could lead to increased interruptions if workloads are shifted to regions with historically higher interruption rates. This could result in suboptimal performance compared to single-region setups, as will be detailed in Section 5.2.4. These findings underscore the importance of employing a careful approach when applying multi-region strategies for spot instances.

To mitigate the risk of increased interruptions, a multi-region strategy should prioritize minimizing disruptions by leveraging insights into the dynamics of "spot instance markets" across different regions. These insights, along with cost and performance optimization goals, should drive "intelligent decision-making" for maximum efficiency and cost-effectiveness. In summary, this motivational study establishes the groundwork by highlighting both the potential benefits and the challenges of multi-region strategies, emphasizing the need for intelligent planning and execution.

3 SPOTVERSE DESIGN

We begin by detailing a design strategy emphasizing key *metrics* such as "Spot Placement Score" and "Interruption Frequency", moving beyond the traditional reliance on spot price history. Next, we delve into the architecture of our proposed SpotVerse system, meticulously developed to manage spot instances effectively. Finally, we highlight the roles of various system components within cloud environments, focusing on deploying heuristic algorithms to efficiently distribute workloads across multiple regions.

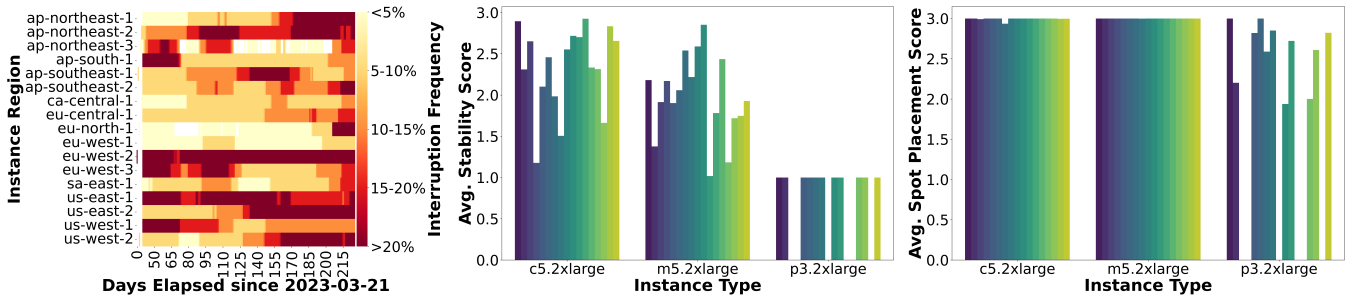
3.1 Utilizing New Metrics for Spot Instance Analysis

As outlined in Section 2, spot instances often suffer from unpredictable interruptions, making them vulnerable. Understanding the "spot instance pricing model" and "spot instance behaviors" is crucial to mitigate unpredictability.

Historically, AWS provided detailed data about spot instance pricing trends, enabling numerous studies focused on better market understanding and developing efficient bidding strategies [78, 88, 105, 107]. However, AWS spot instance policy changed significantly in 2017 [23], leading to more consistent pricing. While this increased price stability, it also weakened the strong correlation between "bid" and "spot" prices. Consequently, new methodologies were needed to leverage spot instance data under the revised policy.

To address these evolving needs, AWS introduced new datasets detailing spot instance dynamics. AWS's "Spot Instance Advisor [37]" now offers detailed metrics for various instance types, including vCPU, memory, savings over on-demand pricing, and interruption frequency. This Interruption Frequency metric quantifies the likelihood of a specific spot instance type being interrupted by AWS. Our study primarily utilizes this Interruption Frequency as a key factor in evaluating the reliability of different spot instance types.

AWS also introduced the "Spot Placement Score" [40], a predictive metric indicating the likelihood of successfully launching a spot instance. This score, ranging from 1 to 10, is influenced by factors such as instance type and region, with scores closer to 10 indicating a higher likelihood of success [41]. Such metrics are crucial for making informed decisions about selecting optimal Availability Zones for specific workloads and enhancing spot capacity utilization. Their significance in guiding decision-making



(a) Interruption Frequency across regions (*m5.2xlarge*)

(b) Average Stability Score across instance types

(c) Average Spot Placement Score across instance types

Figure 4: Metric comparison: Interruption Frequency and Spot Placement Score.

about spot instance usage within cloud environments is highlighted in [85], emphasizing the necessity for adaptability to the dynamic changes in the spot instance market.

Our analysis examined AWS's historical datasets, focusing on "Interruption Frequency" and "Spot Placement Score," as shown in Figure 4. The heatmap in Figure 4a visualizes Interruption Frequency for *m5.2xlarge* instances across regions, with the x-axis representing elapsed time and the y-axis showing regions. The color gradient indicates Interruption Frequency: lighter colors represent rates below 5%, darker shades denote frequencies above 20%, and intermediate colors signify the 5%-20% range. This visualization reveals significant regional variations in Interruption Frequency, underscoring the need for tailored strategies when deploying spot instances in different geographic locations.

Further insights are illustrated in Figures 4b and 4c, which showcase a six-month trajectory of both the Average Interruption Frequency and the Spot Placement Score for *c5.2xlarge*, *m5.2xlarge*, and *p3.2xlarge*. Here, "Average" refers to calculating mean values for each region over six months. The x-axis represents elapsed time from the data collection point, and the primary y-axis corresponds to the Stability Score or Spot Placement Score, depending on the figure. The Stability Score, on a scale from 1 to 3, is inversely proportional to the Interruption Frequency: higher scores indicate a lower probability of interruption, with a score of 3 suggesting an interruption likelihood of less than 5% and a score of 1 indicating a frequency of more than 20%. These figures provide a comprehensive view of the variations in instance behavior and reliability across different regions over time. Notably, specific regions were excluded from the analysis for *p3.2xlarge* instances due to their unavailability in those areas.

Our observations reveal that *c5.2xlarge* and *m5.2xlarge* instances exhibit notable fluctuations in their Average Spot Placement Score across various regions, highlighting the dynamic nature of these instances in different geographical areas. In contrast, *p3.2xlarge* instances display a consistent Spot Placement Score across regions but show variations in their Interruption Frequency. This contrast between the instance types underscores the distinct behaviors of the Interruption Frequency and the Spot Placement Score. AWS datasets offer valuable insights into the spot instance market's historical trends and anticipated directions, which is crucial for planning cloud resources. This data is instrumental in reducing service disruptions and effectively utilizing spot instances to achieve cost savings and enhance infrastructure resilience.

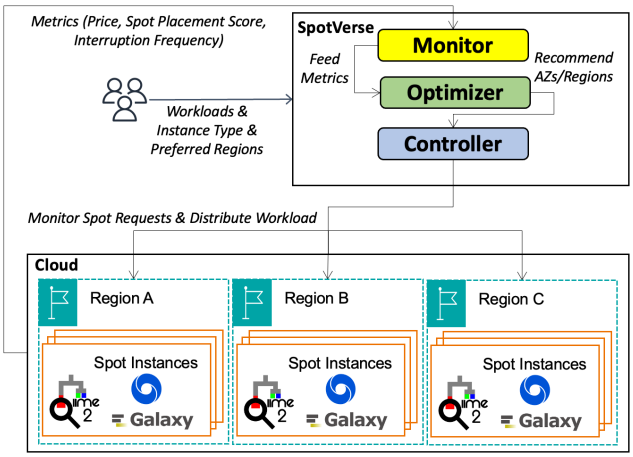


Figure 5: High-level design of SpotVerse.

3.2 Architecture of SpotVerse

Leveraging insights from the latest spot instance metrics, SpotVerse is engineered to pinpoint the "optimal region" for running spot instances, functioning resiliently across multiple regions despite spot interruptions. By introducing intelligent automation and advanced metrics utilization, this approach offers significant advantages over traditional cloud platforms, which typically rely on spot price history and manual configurations. Driven by the input of instance types, a set of workloads, and user-specified regional preferences (or SpotVerse's default recommendations if none are given), this multi-region heuristic approach not only distributes tasks but also provides strategic recommendations, enhancing resource utilization and minimizing costs compared to standard AWS usage. *SpotVerse might even suggest switching to on-demand instances in certain cases*, depending on the preferred regions or instance types, to ensure optimal robustness and cost-effectiveness.

SpotVerse's decision-making process prioritizes both robustness and cost-effectiveness, relying on three key components: **Monitor**, **Controller**, and **Optimizer**. The design employs a heuristic approach that balances cost, reliability, and performance by considering predefined metrics such as Spot Placement Score, Interruption Frequency, and region-specific pricing data, rather than explicit user-specified cost inputs. This approach allows SpotVerse to dynamically adapt to fluctuating cloud conditions and user preferences, indirectly optimizing resource allocation and minimizing costs.

These components carefully select appropriate cloud resources (Spot or On-Demand) in suitable regions, supporting both general and specialized bioinformatics workflows (e.g., those within Galaxy). Figure 5 illustrates the SpotVerse system architecture and the interaction between these core components and the underlying cloud infrastructure.

Monitor: SpotVerse extends AWS CloudWatch's [4] multi-region monitoring capabilities by incorporating "custom rules" tailored for automated spot instance management. These rules enable proactive adjustments and optimizations based on performance and cost-saving metrics. The *Monitor* component further enhances this by providing detailed insights into spot and on-demand prices, Spot Placement Scores, Stability Scores (the inverse of Interruption Frequency), and the status of spot requests (especially pending or unsuccessful ones). This focus on the nuanced dynamics of the spot market, facilitated by the strategic integration with CloudWatch, ensures a more targeted, cost-efficient, and optimized cloud resource utilization strategy across various regions and availability zones.

Optimizer: The *Optimizer* selects optimal regions for executing workloads, utilizing data from the *Monitor*, both when initiating new tasks and responding to interruptions. By introducing automated region selection through a heuristic algorithm based on combined metrics, SpotVerse offers enhanced reliability and cost savings compared to the original cloud platform, which relies on manual region selection.

Controller: The *Controller* receives the resource allocation plan from the *Optimizer* and executes it, ensuring that resources are allocated as designated. By automating the response to spot instance interruptions through workload reallocation, SpotVerse enhances efficiency—a notable improvement over the original cloud platform that typically requires manual intervention. Additionally, the *Controller* handles the initial startup of workloads in optimal regions and initiates a retry mechanism when spot requests fail or remain open, securing necessary resources to ensure workload completion.

3.3 Cloud Region Selection Strategy

The *Optimizer* component of SpotVerse employs a heuristic algorithm (Algorithm 1) to select optimal cloud regions for i) initiating workloads and ii) responding to interruptions. It assesses regions based on a *combined score* derived from the Spot Placement Score and Stability Score, prioritizing regions with higher scores to minimize the risk of interruptions. If the combined score meets or exceeds a predefined threshold, the Optimizer selects the most cost-effective and reliable regions for both scenarios.

Algorithm 1: SpotVerse Workload Management

Input: W: set of workloads;
w: workload experiencing interruption;
I: instance type;
R: maximum number of regions;
T: score threshold.
Initialization:
$S \leftarrow \text{ScoreRegions}(I)$; `// Calculate combined Spot Placement and Stability Scores for all regions`
$V \leftarrow \text{SelectRegions}(S, T)$; `// Filter regions with score` $\geq T$
if $|V| > 0$ **then**
  Sort V by price (ascending);
  $V_{\text{top}} \leftarrow$ Top R regions from V;
  Assign W to regions in V_{top} in a round-robin fashion;
else
  $OD \leftarrow$ CheapestOnDemand();
  Assign W to OD;
On Interruption:
$S \leftarrow \text{ScoreRegions}(I)$; `// Calculate combined Spot Placement and Stability Scores for all regions`
$V \leftarrow \text{SelectRegions}(S, T)$; `// Filter regions with score` $\geq T$
Remove the current interrupted region from V;
if $|V| > 0$ **then**
  Sort V by price (ascending);
  $V_{\text{top}} \leftarrow$ Top R regions from V;
  $r \leftarrow \text{RandomRegion}(V_{\text{top}})$;
  Migrate w to r;
else
  $OD \leftarrow$ CheapestOnDemand();
  Migrate w to OD;

• **For initiating new workloads:** The algorithm first calculates a score for each available region based on the sum of its Spot Placement Score and Stability Score for the specified instance type. It then selects all regions whose scores meet or exceed the predefined threshold. These selected regions are sorted by spot price in ascending order, and the top R regions are chosen for workload distribution. Each workload is then assigned to a region round-robin among the selected regions, optimizing workload distribution and enhancing resource utilization.

• **During interruptions:** The algorithm adapts its selection process by excluding the current region where the interruption occurred. It follows the same scoring and sorting procedure in the initialization phase, selecting the top R regions that meet the threshold. From these selected regions, it randomly picks one to which the interrupted workload is migrated, effectively minimizing disruptions.

On-Demand Instance Strategy: If no regions meet the combined score threshold, suggesting a high interruption risk, SpotVerse activates its on-demand instance strategy. It selects the least

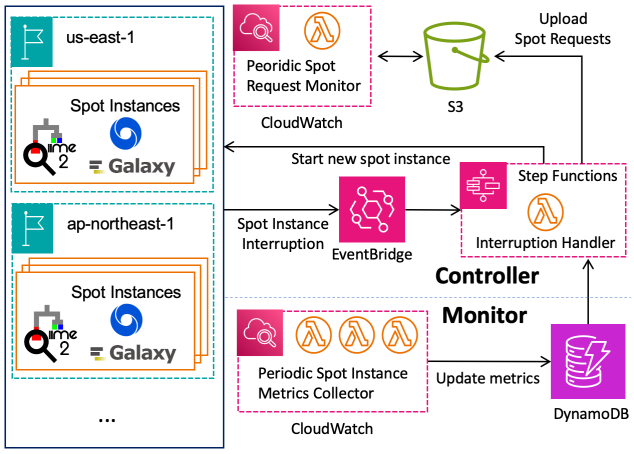


Figure 6: SpotVerse AWS implementation.

expensive on-demand instances across regions to guarantee service continuity. (Further details on this strategy are in Section 5.2.4.

4 SPOTVERSE IMPLEMENTATION

Implementation Overview: Figure 6 illustrates our implementation for managing bioinformatics workloads on AWS spot instances. We leverage AWS Lambda [11], serverless technologies, and AWS CloudFormation [10] to construct a scalable, reliable infrastructure across multiple regions. CloudFormation handles infrastructure deployment, while the AWS SDK [12] manages initial spot instance requests (executed locally). Custom startup scripts initiate and manage bioinformatics workloads within the orchestrated multi-region AWS environment, enabling the execution of any user-defined workload.[1] This allows SpotVerse to support diverse workloads, extending its applicability beyond bioinformatics and enabling scalable management in both general and scientific domains, while its adaptability to other platforms is discussed in Section 7.

Monitor: The Monitor component focuses on monitoring AWS spot instance metrics. We deploy specific Python code and the SpotInfo [42] executable to S3 for general monitoring and Spot Placement Score retrieval, respectively. This is crucial for their utilization within Lambda functions. CloudWatch [4] periodically triggers metrics collector Lambda functions, which systematically gather data, including on-demand/spot prices, Interruption Frequency, and Spot Placement Score. The collected metrics are transmitted to Amazon DynamoDB [5], a centralized data store, enabling our system to handle "real-time" data such as spot fluctuations and interruption notices. This facilitates strategic decision-making regarding instance allocation based on historical trends and performance reliability.

Controller: The Controller manages spot instance requests and handles disruptions. When instances are interrupted (signaled by Amazon EventBridge [33]), the Controller, utilizing the interruption handler Lambda function integrated within Step Functions [34], promptly acquires new instances. Step Functions retry the Lambda function in case of failed or delayed spot requests. Upon initiating a new instance, the Controller uploads relevant details to S3 [8] for monitoring. CloudWatch [4] triggers periodic checks for open requests every 15 minutes, initiating new spot instance requests if necessary. This enhances system robustness and ensures continuous operation. Additionally, as described in Section 3.3, the Controller's Lambda function incorporates the Optimizer's region selection logic, streamlining automation and resource allocation.

Galaxy and Tool Integration: The Galaxy Admin feature [2] is pivotal in integrating essential bioinformatics tools onto a spot instance tailored for Galaxy, offering extensive administrative capabilities. To integrate Galaxy with SpotVerse, this feature is activated by modifying the configuration file, specifically by adding an administrator's email under the `admin_users` parameter, granting control over tool installation and management within Galaxy.

To automate the launch of Galaxy and its workloads at instance startup, a manual setup is performed on an Amazon Machine Image (AMI) [7]. This includes installing and configuring Galaxy (with administrative privileges and an API key), along with necessary tools such as sra-toolkit [43] for extracting SRA data used in workloads [32]. Planemo [60] is integrated for operation initiation, tool downloads, and workload management. Once the customized AMI is created, it is saved and propagated across regions using AWS SDK [12]. Upon instance launch, a user-data script [31], incorporating Planemo and the Galaxy API for running CLI commands, automatically initiates Galaxy and its workloads.

For checkpoint workloads, Planemo[60] initiates execution while DynamoDB[5] updates their status, effectively addressing Galaxy's checkpointing limitations. Activity logs and spot request details are stored in S3 [8] to accurately calculate workload durations and costs. SpotVerse enhances reliability without requiring users to modify their applications. While the initial setup involves manual intervention, future work aims to automate these processes for streamlined integration and operation.

5 EVALUATION

Our evaluation addresses key research questions related to SpotVerse's performance. Specifically, we investigate how SpotVerse enhances cost-effectiveness and resilience in multi-region deployments compared to traditional single-region setups, while also assessing its cost-effectiveness compared to on-demand instances. Additionally, we examine the role of initial workload distributions and thresholds derived from the Spot Placement Score and Interruption Frequency in influencing SpotVerse's overall performance. Our experiments evaluate SpotVerse across various instance types, sizes, and regional deployments, with a focus on key metrics — total completion time, number of interruptions, and overall monetary cost — compared to both traditional single-region and on-demand instances, as well as state-of-the-art frameworks for spot instance management.

5.1 Experimental Setup

5.1.1 Workloads. Our workloads encompass a diverse range of bioinformatics tasks, including both general workflows and those tailored specifically for the Galaxy platform. All workloads are designed to run consistently for 10 to 11 hours. To ensure uniformity

[1] The implementation is available at https://github.com/mjaysonnn/SpotVerse.

across experiments, we strategically incorporate "sleep intervals" into the processing, maintaining consistent duration regardless of the varying specifications of the instances used. Job execution is rigorously managed to operate within operational limits, ensuring interruptions are solely due to preemption, not the exhaustion of hardware resources.

Standard General Workload: QIIME 2 [29], a framework used for in-depth microbiome analysis of DNA sequences [67], facilitates comprehensive studies of microbial communities [49, 66, 69]. This methodology involves several steps: sequence demultiplexing for accurate sample attribution, rigorous quality control with DADA2 [64], phylogenetic tree construction [27], and diversity analysis. Should any interruptions occur, this standard workload necessitates a complete restart.

Galaxy-Specific Workloads: Our galaxy-specific standard workload, the Genome Reconstruction Workload, processes VCF formatted [46] variant datasets from sequenced viral isolates. Each VCF file details nucleotide variations relative to a reference SARS-CoV-2 genome [3]. This 23-step workflow reconstructs viral genomes in FASTA format [17] and classifies them using Pangolin [26]. Any interruption necessitates recomputation of the workload from the beginning.

Our checkpoint workload, the NGS Data Preprocessing Workload, leverages Next Generation Sequencing (NGS) technology [24] to analyze DNA and RNA sequences, a fundamental step in deciphering genetic codes underlying biological mechanisms and growth processes. The NGS data preprocessing workflow encompasses tools such as *FastQC* for early data quality assessment, *MultiQC* for result aggregation, and *Cutadapt*-equivalent methods for sequence trimming. To facilitate checkpointing, we segment the downloaded FastQC dataset and meticulously track each file's processing status. Upon interruption notification, the checkpoint information is updated and uploaded to DynamoDB [5], enabling any new instance to resume processing from the last interruption point seamlessly.

5.1.2 Cloud Setup for Spot Instance Experiments. As outlined in Algorithm 3.3, we set the maximum number of regions to four, prioritizing ease of setup and user convenience. This decision was influenced by the complexity of manually enabling certain regions in AWS [21]. Research suggests that spot instance pricing is not a significant factor [58, 76, 95], so we utilize on-demand pricing for bid prices, with billing occurring at the actual spot price for the specific instance type and region.

For the cost model comparison, we focused on the differential costs incurred by each strategy, including any differences in costs for shared services (like Lambda [11] and CloudWatch [4]), additional services specific to each strategy (like DynamoDB [5] for multi-region), and the total cost of instance usage (accounting for both the base price and any fluctuations due to spot instance pricing). In particular, we accounted for the varying data transfer costs associated with uploading/downloading to S3 [8] for Galaxy-specific checkpoint workloads, recognizing that the multi-region strategy could incur additional costs due to cross-region transfers.

In our experimental environment, the Lambda function was allocated 128MB of memory with a 15-minute timeout limit. We utilized a 1GB dataset of FastQC files from the public Sequence Read Archive (SRA) [32], ensuring upload within the two-minute interruption notice period. Amazon S3 [8] stored details on completed instances and interruptions, while the AWS API [16] provided detailed spot instance pricing for the cost model. Each experiment was conducted three times to account for potential cloud performance and pricing variations.

5.2 Benefits of SpotVerse

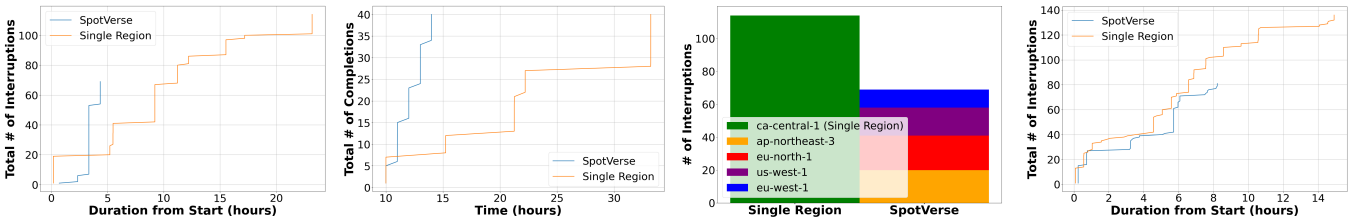
5.2.1 Standard and Checkpoint Workloads. A comprehensive set of experiments evaluated SpotVerse's benefits, independent of specific workload types. We began in the *ca-central-1* region, chosen for its cost-effectiveness for *m5.xlarge* instances. The single-region (baseline) approach initiates instances exclusively in this region, while SpotVerse starts there but subsequently migrates instances based on Algorithm 1, excluding the initial distribution strategy for fair comparison. The impact of the initial distribution strategy is discussed in Section 5.2.3. Two types of Galaxy-specific workloads, *standard* and *checkpoint* (detailed in Section 5.1.1), were run with 40 parallel instances per experiment. The assessment focused on total completion time, interruption details, monetary cost, and regional distribution per strategy.

Figure 7 details the performance of standard and checkpoint workloads under single-region and multi-region approaches using SpotVerse. Figures 7a and 7d plot workload duration (x-axis) against cumulative interruptions (y-axis). In Figure 7b, the x-axis tracks the elapsed time since the start of each workload execution, and the y-axis shows the number of instances completing tasks. Figure 7c illustrates the regional distribution of interruptions for standard workloads under both strategies (The regional distribution for checkpoint workloads is similar and thus omitted for conciseness.)

SpotVerse dramatically improves efficiency and cost savings for standard workloads, as shown in Figures 7a and 7b. It reduces total completion time from approximately 33 to 14 hours and interruption count by nearly 39% (114 to 69). This translates to a cost of $41.46, considerably lower than the single region's $73.92 and the on-demand cost of $77.81. These findings demonstrate that choosing the cheapest region may *not* always yield the most cost-effective results, highlighting SpotVerse's advantages over traditional spot instance strategies, particularly for fault-tolerant (checkpoint) workloads [6]. Given that the duration of on-demand instances is 10 to 11 hours, SpotVerse achieves a comparable duration of 14 hours with a significantly reduced cost (46.7% reduction), further demonstrating its efficiency and cost savings.

The bar chart in Figure 7c illustrates the regional distribution of interruptions for standard workloads under two different strategies: single-region and SpotVerse. The single-region strategy relies solely on the *ca-central-1* region (solid green bar), where a notable number of interruptions occur. In contrast, SpotVerse's multi-region strategy effectively reduces the number of spot instance interruptions. This is depicted by the stacked bar, where each color represents a different region and its respective interruption count. This visual comparison underscores SpotVerse's ability to diversify risk and enhance reliability by utilizing multiple regions.

As spot instances are suitable for fault-tolerant workloads, they offer cost benefits compared to on-demand instances [50]. However,



(a) Cumulative interruptions for standard workload

(b) Total completion time for standard workload

(c) Interruption distribution for standard workload

(d) Cumulative interruptions for checkpoint workload

Figure 7: Performance comparison: SpotVerse vs. single-region deployment for different workloads.

SpotVerse further enhances these benefits by reducing both costs and interruptions. Figure 7d compares the number of interruptions for checkpoint workloads between the Single Region approach and SpotVerse. SpotVerse reduces interruptions by approximately 40% (136 to 81) and cuts costs by about 11% (from $29.64 to $26.26). While SpotVerse also improves total completion time (from 15.46 to 11.75 hours), this is not shown in a separate graph due to space constraints.

Instance Type	Baseline Region
m5.large	us-west-2
m5.xlarge	ca-central-1
m5.2xlarge	ap-northeast-3
r5.2xlarge	ca-central-1
c5.2xlarge	eu-north-1

Table 1: Baseline regions for various spot instance types.

5.2.2 Instance Types, Sizes, and Specifications. We conducted experiments focusing on the *standard* general workload, detailed in Section 5.1.1, using 40 instances across various specifications to assess SpotVerse's performance under diverse operational conditions. These conditions included different instance types and sizes, with baseline configurations detailed in Table 1, chosen for their cost-effectiveness on the experiment date. We systematically compared the number of interruptions, total completion time, and costs to gauge performance and cost efficiency.

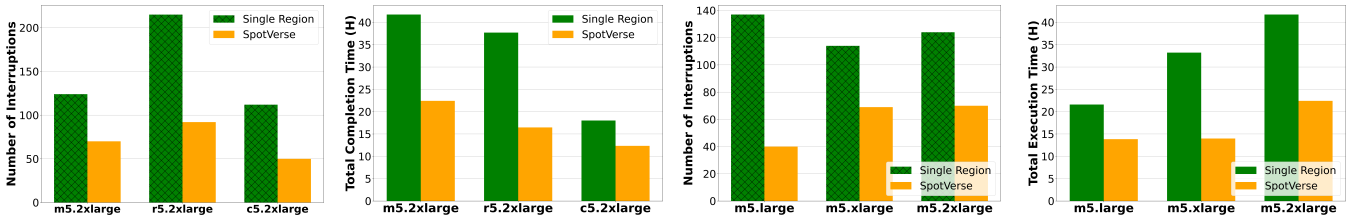
In our comparative analysis (shown in Figures 8a and 8b), SpotVerse significantly enhanced the performance of AWS instances across different types with similar specifications. Notably, *r5.2xlarge* instances in the baseline region (with the lowest Stability Score of 1) saw the most substantial reduction in interruptions, decreasing from 215 to 92, leading to significantly shorter completion times. This demonstrates SpotVerse's effectiveness in high-risk setups, achieving cost savings of nearly 52% and reducing completion times by approximately 56%. Additionally, *c5.2xlarge* instances demonstrated the highest cost reduction (52%) compared to the on-demand instances. Note that the Stability Score is *inversely proportional* to the Interruption Frequency, meaning that the regions with lower scores typically experience higher interruption rates.

Figures 8c and 8d further highlight SpotVerse's performance advantages over single-region deployments, focusing on various sizes of the same instance family (*m5*). Notably, the *m5.large* instances, with the lowest Stability Score of 1, showed significant reductions in interruptions (from 137 to 40) and a cost reduction of approximately 27% (from $41.7 to $29.1). Additionally, when compared to on-demand instances, the *m5.xlarge* instance type, leveraging SpotVerse's multi-region strategy, achieved the most significant cost reduction, reaching up to 47%. These results clearly demonstrate SpotVerse's capacity to optimize performance across different sizes within the same instance family and across AWS instances with similar specifications.

5.2.3 Initial Workload Distribution Strategy. To demonstrate the influence of the initial distribution region strategy (part of the heuristic algorithm described in Section 3.3), we conducted experiments using *m5.xlarge* instances with a *standard* Galaxy-specific workload (detailed in Section 5.1.1) across 40 instances. For the baseline, we selected *ap-northeast-3* due to its highest combined Stability and Spot Placement Scores, along with the lowest spot price among similarly high-scoring regions. For SpotVerse, we chose four top scoring regions: *ap-northeast-3*, *eu-north-1*, *us-west-1*, and *eu-west-1*. SpotVerse initiated operations in these regions and transitioned to one of them upon interruption. We compared the number of interruptions, total completion time, and monetary costs for both approaches.

Figure 9 illustrates the impact of the initial regional distribution strategy on SpotVerse's performance. Distributing workloads across recommended regions significantly reduces the number of interruptions for both Standard and Checkpoint workloads, particularly for the Standard workload where interruptions decrease by approximately 32% (from 69 to 42), as shown in Figure 9a. Moreover, Figure 9b demonstrates the efficiency and cost-effectiveness of SpotVerse's initial distribution strategy, leading to a reduction of up to 12% in total completion time and 11% in monetary costs for both workload types.

5.2.4 Evaluating SpotVerse's Threshold-Based Allocation. The cloud resource allocation strategy of SpotVerse (Algorithm 1) utilizes thresholds based on the combined Spot Placement Score and Stability Score. Regions with high scores are preferred for cost-effective workload allocation on spot instances, while low-scoring regions, indicating lower reliability, favor on-demand instances. Our experiments, conducted across regions with varying scores, assess how these threshold settings impact SpotVerse's performance, especially when on-demand instances offer better reliability.



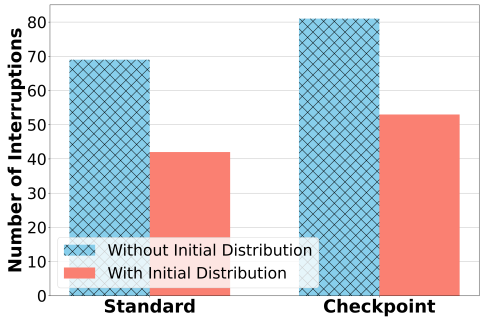
(a) Number of interruptions by instance type

(b) Completion time by instance type

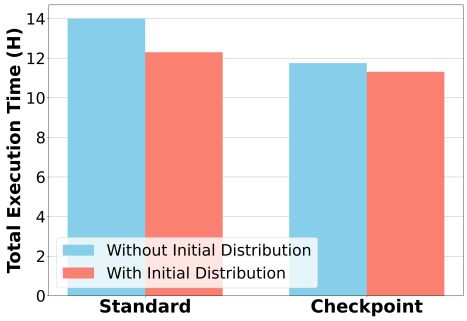
(c) Number of interruptions by instance size

(d) Completion time by instance size

Figure 8: Performance impact of instance types and sizes: Number of interruptions and completion times.



(a) Cumulative interruptions by workload type



(b) Total completion time by workload type

Figure 9: Impact of the initial regional distribution strategy on SpotVerse's performance across different workload types.

Experiment Setting	Values
Duration (Hours)	5, 10, 20
Threshold	4, 5, 6

Table 2: Configurations for evaluating the impact of thresholds on SpotVerse's performance.

We used various threshold settings (4, 5, and 6) and workload durations (5, 10, and 20 hours) for our experiments with *m5.xlarge* instances and the standard general workload (Section 5.1.1). Table 2 details these configurations, while Table 3 shows the four regions selected for each threshold, prioritizing those with higher

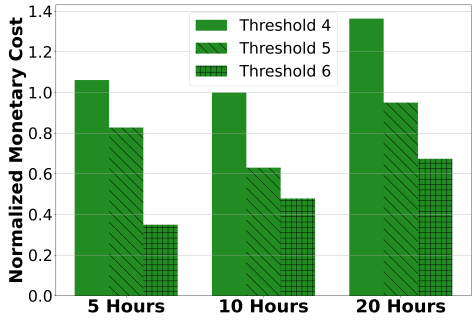


Figure 10: Normalized cost of *m5.xlarge* instances (varying durations) under different thresholds, relative to the cheapest on-demand instances.

Threshold	Selected Regions
6	us-west-1, ap-northeast-3, eu-west-1, eu-north-1
5	ap-southeast-1, eu-west-3, ca-central-1, eu-west-2
4	us-east-1, us-east-2, ap-southeast-2, us-west-2

Table 3: Regions selected for each threshold value.

combined scores. Note that threshold 4's chosen regions were the cheapest among all regions. Figure 10 illustrates the relative cost-effectiveness of spot instances compared to on-demand instances using the "normalized" monetary cost (calculated by running workloads in the cheapest region for on-demand instances). The values below 1 indicate cost savings, whereas the values above 1 indicate that the spot instances are more expensive.

Thresholds 5 and 6 consistently demonstrate cost savings across durations (up to 65% compared to on-demand), highlighting SpotVerse's effectiveness. However, focusing solely on cost (threshold 4) can lead to increased interruptions and a cost increase of up to 36%. This emphasizes the importance of SpotVerse's balanced approach, considering cost and reliability. SpotVerse recommends on-demand instances in user-preferred regions when necessary, demonstrating

its adaptability. Notably, as duration increases, SpotVerse's cost-savings over on-demand instances diminish, highlighting the need to address the limitations of spot instances for extended durations.

	Number of interruptions	Cost ($)	Completion time (hours)
SpotVerse	42	36.73	12.3
SkyPilot	129	74.76	30.9

Table 4: Comparison of SpotVerse and SkyPilot in terms of the number of interruptions, cost, and completion time.

5.2.5 Comparison Against State-of-the-Art Frameworks. To benchmark SpotVerse against other leading frameworks, we conducted a comparative study with SkyPilot [104]. SkyPilot is designed to run large-scale computations, including Large Language Models (LLMs) and batch jobs, across multiple clouds/regions. It leverages spot instances for cost savings and high availability, automating the search for the least expensive resources and managing the job life-cycle across regions, automatically relaunching interrupted tasks due to spot instance preemptions.

For the experiments, both frameworks utilized a standard general workload (detailed in Section 5.2.1) consisting of 40 instances running for 10 to 11 hours, with configurations set to automatically restart jobs upon interruption. To ensure a fair comparison, SkyPilot's YAML configuration file [35] was adjusted to mirror the command jobs used in SpotVerse workflows. Performance metrics, including number of interruptions, total completion time, and monetary cost, were tracked using Amazon S3 [8], which logs detailed instance usage and interruption data.

The results, shown in Table 4, indicate that SpotVerse significantly reduces costs and improves completion times compared to SkyPilot. SpotVerse achieves a 51% cost reduction (from $74.76 to $36.73) and a 60% reduction in completion time (from 30.9 hours to 12.3 hours), showcasing its efficiency. While SkyPilot aims to minimize costs by searching for the cheapest resources across regions, it is less consistent and experiences more operational disruptions due to its interruption handling, as acknowledged in its documentation. These findings highlight SpotVerse's superior strategy, which *balances* cost and reliability, proving to be a more effective solution for cloud resource management.

6 RELATED WORK

Cloud Computing in Bioinformatics: The computational demands of bioinformatics necessitate the use of cloud computing, providing the flexibility, scalability, and accessibility essential for modern research [56, 65, 75, 86, 98]. Numerous frameworks, such as QIIME 2 for microbial ecology, GATK for variant discovery, and AlphaFold for protein structure prediction, have been developed to support bioinformatics workloads [18, 28, 59]. Integrating platforms like Galaxy with cloud environments have significantly enhanced scalability and computational efficiency [74, 89, 92, 96]. Kubernetes-based approaches for streamlined workflow execution in cloud ecosystems have yielded notable success [91], while tools such as Gyan [74] have facilitated GPU-aware mapping and orchestration of Galaxy tools on GPU-equipped cloud clusters. SpotVerse builds upon these advancements by optimizing the cost-effective use of spot instances across regions, further promoting cost optimization for bioinformatics workflows.

Multi-Region Strategies and Optimization Frameworks for Spot Instances: Recent research has explored various methodologies to manage spot instances effectively across multiple regions and clouds [53, 54, 77, 79, 84, 85, 87, 93, 103, 104, 106]. SkyPilot [104] and DeepSpotCloud [84] have demonstrated the advantages of multi-region spot instance deployment in bioinformatics and deep learning, respectively. While these studies highlight potential efficiencies and cost savings, strategies prioritizing the lowest price alone can increase the risk of interruptions. In contrast, SpotVerse advances its multi-region strategy by prioritizing Interruption Frequency and Spot Placement Score, favoring regions with the lowest preemption rates to enhance reliability. Unlike prior frameworks focusing primarily on cost, SpotVerse strategically balances workload distribution across optimally-selected regions, improving *both* cost-effectiveness and performance.

Spot Instance Availability Data: Understanding spot instance availability is vital for optimizing cloud costs. Research by Lee et al. [85] introduced metrics that combine spot instance availability with Interruption Frequency data, significantly enhancing resource management in spot markets. Similarly, Pham et al. [95] utilized Interruption Frequency data to further enrich the analytical framework of the spot market. Leveraging the comprehensive dataset and metrics from SpotLake [85], SpotVerse optimizes computational resource allocation by dynamically switching between on-demand and spot market pricing to enhance stability. This strategic distribution of tasks across regions minimizes interruption risks and is particularly advantageous in bioinformatics, where tasks often have long durations. By adeptly navigating spot market volatility, SpotVerse creates a resilient framework for cloud resource optimization.

7 FUTURE WORK

We plan to investigate how resource usage impacts spot instance interruptions depending on the day or time of the week, as we have observed differences in these patterns during our experiments. Specifically, we plan to use machine learning to optimize cloud resource allocation, predict efficient resource configurations, and adapt to market conditions, ultimately enhancing cost-effectiveness and operational efficiency.

SpotVerse currently utilizes Amazon S3 for uploading and downloading data. However, the two-minute interruption notice period of spot instances and S3's limitations regarding large data transfers pose challenges. In future work, we plan to explore alternative storage solutions such as Elastic File System (EFS) [45] and other advanced storage technologies to mitigate the challenges associated with large data sizes and inter-region transfers. This would enhance the feasibility and efficiency of operating in a multi-region environment, particularly for workloads involving substantial volumes.

While SpotVerse is initially built on AWS CloudFormation [10], a robust infrastructure management platform, the global trend towards multi-cloud strategies necessitates adaptability. Other cloud providers offer similar services, such as Azure Resource Manager [13]

and Google Cloud Deployment Manager [19], which could facilitate SpotVerse's adaptation to these platforms.

However, adapting to different providers introduces challenges due to the variability in available metrics, such as Interruption Frequency and Spot Placement Score. For instance, Azure only provides Interruption Frequency data, while Google Cloud Platform (GCP) currently lacks comprehensive spot instance metrics. Addressing these variability and compatibility issues is crucial for leveraging the distinct advantages offered by each cloud service. Additionally, we aim to develop an intuitive user interface and a universal API to simplify interactions and enhance user control and visibility within SpotVerse.

8 CONCLUSIONS

SpotVerse is a cloud software designed to manage various types of workloads, including the bioinformatics workflows from the Galaxy framework, across multiple regions. It tackles the challenge of spot instance interruptions through intelligent resource allocation and dynamic switching between spot and on-demand instances, optimizing cost and job reliability. By considering factors such as interruption frequency and spot placement score, SpotVerse achieves significant cost reductions (up to 52%) compared to the traditional methods. Although validated primarily with bioinformatics workloads, SpotVerse's effectiveness as a robust and cost-efficient solution for executing large-scale workloads in the cloud extends to a wide range of applications beyond this domain.

ACKNOWLEDGMENTS

We thank the Middleware reviewers, Jisoo Min, and our shepherd, Christine Julien, for their valuable feedback. This work is supported in part by NSF grants 1931531, 2149389, and 2122155.

REFERENCES

[1] 2022. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2022 update. *Nucleic Acids Research* 50, W1 (2022), W345–W351.
[2] 2023. Galaxy Administration. https://galaxyproject.org/admin/.
[3] 2023. SARS-CoV-2 lineage assignment. https://training.galaxyproject.org/training-material/topics/galaxy-interface/tutorials/workflow-automation/tutorial.html.
[4] 2024. Amazon CloudWatch. https://aws.amazon.com/cloudwatch/.
[5] 2024. Amazon DynamoDB. https://aws.amazon.com/dynamodb/.
[6] 2024. Amazon EC2 Spot Instances. https://aws.amazon.com/ec2/spot/.
[7] 2024. Amazon Machine Images (AMI). https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html.
[8] 2024. Amazon Simple Storage Service (Amazon S3). https://aws.amazon.com/s3/.
[9] 2024. Amazon Web Services. https://aws.amazon.com/.
[10] 2024. AWS CloudFormation. https://aws.amazon.com/cloudformation/.
[11] 2024. AWS Lambda. https://aws.amazon.com/lambda/.
[12] 2024. AWS SDK for Python (Boto3). https://boto3.amazonaws.com/v1/documentation/api/latest/index.html.
[13] 2024. Azure Resource Manager. https://azure.microsoft.com/en-us/get-started/azure-portal/resource-manager/.
[14] 2024. Azure Spot Virtual Machines. https://azure.microsoft.com/en-us/pricing/spot/.
[15] 2024. DeepVariant: A Highly Accurate Genetic Variant Caller Using Deep Neural Networks. https://github.com/google/deepvariant.
[16] 2024. Describe Spot Price History. https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-spot-price-history.html.
[17] 2024. FASTA Format for Nucleotide Sequences. https://www.ncbi.nlm.nih.gov/genbank/fastaformat/.
[18] 2024. Genome Analysis Toolkit. https://gatk.broadinstitute.org/hc/en-us.
[19] 2024. Google Cloud Deployment Manager. https://www.jic.ac.uk/blog/what-is-microbial-science/.
[20] 2024. Google Cloud Platform. https://cloud.google.com/.
[21] 2024. Managing Account Regions. https://docs.aws.amazon.com/accounts/latest/reference/manage-acct-regions.html.
[22] 2024. Microsoft Azure. https://azure.microsoft.com/en-us.
[23] 2024. New Amazon EC2 Spot pricing model. https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing/.
[24] 2024. Next-Generation Sequencing Data Analysis. https://www.ecseq.com/support/ngs/getting-started-with-ngs-data-analysis-overview.
[25] 2024. On-Demand Instances. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-on-demand-instances.html.
[26] 2024. Phylogenetic Assignment of Named Global Outbreak LINeages. https://github.com/cov-lineages/pangolin.
[27] 2024. Phylogenetic Tree. https://en.wikipedia.org/wiki/Phylogenetic_tree.
[28] 2024. protein structure prediction with AlphaFold. https://deepmind.google/technologies/alphafold/.
[29] 2024. QIIME 2: Next-Generation Microbial Community Analysis. https://qiime2.org/.
[30] 2024. Regions and Availability Zones. https://aws.amazon.com/about-aws/global-infrastructure/regions_az/.
[31] 2024. Run commands on your Linux instance at launch. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html.
[32] 2024. Sequence Read Archive. https://www.ncbi.nlm.nih.gov/sra.
[33] 2024. Serverless Event Router – Amazon EventBridge. https://aws.amazon.com/eventbridge/.
[34] 2024. Serverless Workflow Orchestration – AWS Step Functions. https://aws.amazon.com/step-functions/.
[35] 2024. SkyPilot Task YAML. https://skypilot.readthedocs.io/en/latest/reference/yaml-spec.html.
[36] 2024. Software-as-a-Service (SaaS) on AWS. https://aws.amazon.com/solutions/saas/.
[37] 2024. Spot Instance advisor. https://aws.amazon.com/ec2/spot/instance-advisor/.
[38] 2024. Spot Instance interruption notices. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-instance-termination-notices.html.
[39] 2024. Spot Instance interruptions. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html.
[40] 2024. Spot Placement Score. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-placement-score.html.
[41] 2024. Spot request status. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-request-status.html.
[42] 2024. SpotInfo Tool. https://github.com/alexei-led/spotinfo.
[43] 2024. SRA Toolkit. https://hpc.nih.gov/apps/sratoolkit.html.
[44] 2024. TensorFlow Bioinformatics: Integrating TensorFlow for Advanced Genomic Analysis. https://www.tensorflow.org/io/tutorials/genome.
[45] 2024. Use Amazon EFS with Amazon EC2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEFS.html.
[46] 2024. Variant Call Format. https://samtools.github.io/hts-specs/VCFv4.2.pdf.
[47] 2024. What is FaaS (Function-as-a-Service)? https://www.ibm.com/topics/faas.
[48] 2024. What is IaaS (Infrastructure as a Service)? https://aws.amazon.com/what-is/iaas/.
[49] 2024. What is microbial science? https://www.jic.ac.uk/blog/what-is-microbial-science/.
[50] 2024. When to use spot instances. https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-leveraging-ec2-spot-instances.
[51] Enis Afgan, Dannon Baker, Nate Coraor, Hiroki Goto, Ian M Paul, Kateryna D Makova, Anton Nekrutenko, and James Taylor. 2011. Harnessing cloud computing with Galaxy Cloud. *Nature biotechnology* 29, 11 (2011), 972–974.
[52] Enis Afgan, Brad Chapman, Margita Jadan, Vedran Franke, and James Taylor. 2012. Using cloud computing infrastructure with CloudBioLinux, CloudMan, and Galaxy. *Current protocols in bioinformatics* 38, 1 (2012), 11–9.
[53] Austin Aske and Xinghui Zhao. 2018. Supporting multi-provider serverless computing on the edge. In *Workshop Proceedings of the 47th International Conference on Parallel Processing*. 1–6.
[54] Ataollah Fatahi Baarzi, George Kesidis, Carlee Joe-Wong, and Mohammad Shahrad. 2021. On merits and viability of multi-cloud serverless. In *Proceedings of the ACM Symposium on Cloud Computing*. 600–608.
[55] Qanita Bani Baker, Mahmoud Hammad, Wesam Al-Rashdan, Yaser Jararweh, AL-Smadi Mohammad, and Mohammad Al-Zinati. 2020. Comprehensive comparison of cloud-based NGS data analysis and alignment tools. *Informatics in Medicine Unlocked* 18 (2020), 100296.
[56] Bayan H Banimfreg. 2023. A comprehensive review and conceptual framework for cloud computing adoption in bioinformatics. *Healthcare Analytics* (2023), 100190.
[57] Chris Barnett. 2020. Galaxy Project: Cheminformatics and Computational Chemistry. (2 2020). https://doi.org/10.25375/uct.11912586.v1
[58] Matt Baughman, Simon Caton, Christian Haas, Ryan Chard, Rich Wolski, Ian Foster, and Kyle Chard. 2019. Deconstructing the 2017 changes to AWS spot market pricing. In *Proceedings of the 10th Workshop on Scientific Cloud Computing*. 19–26.

[59] Evan Bolyen, Jai Ram Rideout, Matthew R Dillon, Nicholas A Bokulich, Christian Abnet, Gabriel A Al-Ghalith, Harriet Alexander, Eric J Alm, Manimozhiyan Arumugam, Francesco Asnicar, et al. 2018. *QIIME 2: Reproducible, interactive, scalable, and extensible microbiome data science.* Technical Report. PeerJ Preprints.

[60] Simon Bray, John Chilton, Matthias Bernt, Nicola Soranzo, Marius van den Beek, Bérénice Batut, Helena Rasche, Martin Čech, Peter JA Cock, Björn Grüning, et al. 2023. The Planemo toolkit for developing, deploying, and executing scientific data analyses in Galaxy and beyond. *Genome Research* 33, 2 (2023), 261–268.

[61] Simon Bray, Tim Dudgeon, Rachael Skyner, Rolf Backofen, Björn Grüning, and Frank von Delft. 2022. Galaxy workflows for fragment-based virtual screening: a case study on the SARS-CoV-2 main protease. *Journal of Cheminformatics* 14, 1 (2022), 1–13.

[62] Simon A. Bray, Xavier Lucas, Anup Kumar, and Björn A. Grüning. 2020. The ChemicalToolbox: reproducible, user-friendly cheminformatics analysis on the Galaxy platform. *Journal of Cheminformatics* 12, 1 (jun 2020). https://doi.org/10.1186/s13321-020-00442-7

[63] Brendan Burns, Brian Grant, David Oppenheimer, Eric Brewer, and John Wilkes. 2016. Borg, omega, and kubernetes. *Commun. ACM* 59, 5 (2016), 50–57.

[64] Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. 2016. DADA2: High-resolution sample inference from Illumina amplicon data. *Nature methods* 13, 7 (2016), 581–583.

[65] Christiam Camacho, Grzegorz M Boratyn, Victor Joukov, Roberto Vera Alvarez, and Thomas L Madden. 2023. ElasticBLAST: accelerating sequence search via cloud computing. *BMC bioinformatics* 24, 1 (2023), 1–16.

[66] J Gregory Caporaso, Christian L Lauber, Elizabeth K Costello, Donna Berg-Lyons, Antonio Gonzalez, Jesse Stombaugh, Dan Knights, Pawel Gajer, Jacques Ravel, Noah Fierer, et al. 2011. Moving pictures of the human microbiome. *Genome biology* 12 (2011), 1–8.

[67] Marcus J Claesson, Adam G Clooney, and Paul W O'toole. 2017. A clinician's guide to microbiome analysis. *Nature Reviews Gastroenterology & Hepatology* 14, 10 (2017), 585–595.

[68] Alfonso Esposito, Chiara Colantuono, Valentino Ruggieri, and Maria Luisa Chiusano. 2016. Bioinformatics for agriculture in the next-generation sequencing era. *Chemical and Biological Technologies in Agriculture* 3 (2016), 1–12.

[69] Jack A Gilbert, Janet K Jansson, and Rob Knight. 2014. The Earth Microbiome project: successes and aspirations. *BMC biology* 12 (2014), 1–4.

[70] Gloria I Giraldo-Calderón, Omar S Harb, Sarah A Kelly, Samuel SC Rund, David S Roos, and Mary Ann McDowell. 2022. VectorBase. org updates: bioinformatic resources for invertebrate vectors of human pathogens and related organisms. *Current opinion in insect science* 50 (2022), 100860.

[71] Ilias Glogovitis, Galina Yahubyan, Thomas Würdinger, Danijela Koppers-Lalic, and Vesselin Baev. 2021. MiRGalaxy: Galaxy-Based Framework for Interactive Analysis of MicroRNA and IsomiR Sequencing Data. *Cancers* 13, 22 (2021), 5663.

[72] Google Cloud. 2024. Preemptible Virtual Machines. https://cloud.google.com/preemptible-vms.

[73] Qiang Gu, Anup Kumar, Simon Bray, Allison Creason, Alireza Khanteymoori, Vahid Jalili, Björn Grüning, and Jeremy Goecks. 2021. Galaxy-ML: An accessible, reproducible, and scalable machine learning toolkit for biomedicine. *PLoS computational biology* 17, 6 (2021), e1009014.

[74] Gulsum Gudukbay, Jashwant Raj Gunasekaran, Yilin Feng, Mahmut T Kandemir, Anton Nekrutenko, Chita R Das, Paul Medvedev, Björn Gruning, Nate Coraor, Nathan Roach, et al. 2021. GYAN: Accelerating bioinformatics tools in galaxy with GPU-aware computation mapping. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 194–203.

[75] Naiyar Iqbal and Pradeep Kumar. 2023. From Data Science to Bioscience: Emerging era of bioinformatics applications, tools and challenges. *Procedia Computer Science* 218 (2023), 1516–1528.

[76] David Irwin, Prashant Shenoy, Pradeep Ambati, Prateek Sharma, Supreeth Shastri, and Ahmed Ali-Eldin. 2019. The price is (not) right: Reflections on pricing for transient cloud servers. In *2019 28th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 1–9.

[77] Paras Jain, Sam Kumar, Sarah Wooders, Shishir G Patil, Joseph E Gonzalez, and Ion Stoica. 2023. Skyplane: Optimizing Transfer Cost and Throughput Using {Cloud-Aware} Overlays. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1375–1389.

[78] Mikhail Khodak, Liang Zheng, Andrew S Lan, Carlee Joe-Wong, and Mung Chiang. 2018. Learning cloud dynamics to optimize spot instance bidding strategies. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2762–2770.

[79] Kyunghwan Kim, Subin Park, Jaeil Hwang, Hyeonyoung Lee, Seokhyeon Kang, and Kyungyong Lee. 2023. Public Spot Instance Dataset Archive Service. In *Companion Proceedings of the ACM Web Conference 2023*. 69–72.

[80] Konstantinos Krampis, Tim Booth, Brad Chapman, Bela Tiwari, Mesude Bicak, Dawn Field, and Karen E Nelson. 2012. Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. *BMC bioinformatics* 13 (2012), 1–8.

[81] Priyanka Kumari and Yogesh Kumar. 2021. Bioinformatics and computational tools in bioremediation and biodegradation of environmental pollutants. In *Bioremediation for environmental sustainability*. Elsevier, 421–444.

[82] Gregory M Kurtzer, Vanessa Sochat, and Michael W Bauer. 2017. Singularity: Scientific containers for mobility of compute. *PloS one* 12, 5 (2017), e0177459.

[83] Ben Langmead and Abhinav Nellore. 2018. Cloud computing for genomic data analysis and collaboration. *Nature Reviews Genetics* 19, 4 (2018), 208–219.

[84] Kyungyong Lee and Myungjun Son. 2017. Deepspotcloud: leveraging cross-region gpu spot instances for deep learning. In *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE, 98–105.

[85] Sungjae Lee, Jaeil Hwang, and Kyungyong Lee. 2022. Spotlake: Diverse spot instance dataset archive service. In *2022 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 242–255.

[86] Kevin G Libuit, Emma L Doughty, James R Otieno, Frank Ambrosio, Curtis J Kapsak, Emily A Smith, Sage M Wright, Michelle R Scribner, Robert A Petit III III, Catarina Inês Mendes, et al. 2023. Accelerating bioinformatics implementation in public health. *Microbial Genomics* 9, 7 (2023), 001051.

[87] Wei-Tsung Lin, Chandra Krintz, and Rich Wolski. 2018. Tracing function dependencies across clouds. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 253–260.

[88] Markus Lumpe, Mohan Baruwal Chhetri, Quoc Bao Vo, and Ryszard Kowalcyk. 2017. On estimating minimum bids for Amazon EC2 spot instances. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 391–400.

[89] Antonio Maciá-Lillo, Tamai Ramírez, Higinio Mora, Antonio Jimeno-Morenilla, and José-Luis Sánchez-Romero. 2023. GPU Cloud Architectures for Bioinformatic Applications. In *International Work-Conference on Bioinformatics and Biomedical Engineering*. Springer, 77–89.

[90] Dirk Merkel et al. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux j* 239, 2 (2014), 2.

[91] Pablo Moreno, Luca Pireddu, Pierrick Roger, Nuwan Goonasekera, Enis Afgan, Marius Van Den Beek, Sijin He, Anders Larsson, Daniel Schober, Christoph Ruttkies, et al. 2018. Galaxy-Kubernetes integration: scaling bioinformatics workflows in the cloud. *BioRxiv* (2018), 488643.

[92] Theodore M Nelson, Sankar Ghosh, and Thomas S Postler. 2022. L-RAPiT: A Cloud-Based Computing Pipeline for the Analysis of Long-Read RNA Sequencing Data. *International Journal of Molecular Sciences* 23, 24 (2022), 15851.

[93] Hai Duc Nguyen and Andrew A Chien. 2023. Storm-RTS: Stream Processing with Stable Performance for Multi-cloud and Cloud-edge. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. IEEE, 45–57.

[94] Yuriy L Orlov, Anastasia A Anashkina, Vadim V Klimontov, and Ancha V Baranova. 2021. Medical genetics, genomics and bioinformatics aid in understanding molecular mechanisms of human diseases. , 9962 pages.

[95] Thanh-Phuong Pham, Sasko Ristov, and Thomas Fahringer. 2018. Performance and behavior characterization of amazon ec2 spot instances. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 73–81.

[96] Niko Pinter, Damian Glätzer, Matthias Fahrner, Klemens Fröhlich, James Johnson, Björn Andreas Grüning, Bettina Warscheid, Friedel Drepper, Oliver Schilling, and Melanie Christine Föll. 2022. MaxQuant and MSstats in galaxy enable reproducible cloud-based analysis of quantitative proteomics experiments for everyone. *Journal of Proteome Research* 21, 6 (2022), 1558–1565.

[97] Xu-Bo Qian, Tong Chen, Yi-Ping Xu, Lei Chen, Fu-Xiang Sun, Mei-Ping Lu, and Yong-Xin Liu. 2020. A guide to human microbiome research: study design, sample collection, and bioinformatics analysis. *Chinese Medical Journal* 133, 15 (2020), 1844–1855.

[98] Jinlong Ru, Mohammadali Khan Mirzaei, Jinling Xue, Xue Peng, and Li Deng. 2023. ViroProfiler: a containerized bioinformatics pipeline for viral metagenomic data analysis. *Gut Microbes* 15, 1 (2023), 2192522.

[99] Denis Schapiro, Artem Sokolov, Clarence Yapp, Yu-An Chen, Jeremy L Muhlich, Joshua Hess, Allison L Creason, Ajit J Nirmal, Gregory J Baker, Maulik K Nariya, et al. 2022. MCMICRO: a scalable, modular image-processing pipeline for multiplexed tissue imaging. *Nature methods* 19, 3 (2022), 311–315.

[100] Jay Shendure, Shankar Balasubramanian, George M Church, Walter Gilbert, Jane Rogers, Jeffery A Schloss, and Robert H Waterston. 2017. DNA sequencing at 40: past, present and future. *Nature* 550, 7676 (2017), 345–353.

[101] Marco Antonio Tangaro, Pietro Mandreoli, Matteo Chiara, Giacinto Donvito, Marica Antonacci, Antonio Parisi, Angelica Bianco, Angelo Romano, Daniela Manila Bianchi, Davide Cangelosi, et al. 2021. Laniakea@ ReCaS: exploring the potential of customisable Galaxy on-demand instances as a cloud-based service. *BMC bioinformatics* 22, 15 (2021), 1–21.

[102] Jared Wilkening, Andreas Wilke, Narayan Desai, and Folker Meyer. 2009. Using clouds for metagenomics: a case study. In *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, 1–6.

[103] Zhanghao Wu, Wei-Lin Chiang, Ziming Mao, Zongheng Yang, Eric Friedman, Scott Shenker, and Ion Stoica. 2024. Can't Be Late: Optimizing Spot Instance Savings under Deadlines. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. 185–203.

[104] Zongheng Yang, Zhanghao Wu, Michael Luo, Wei-Lin Chiang, Romil Bhardwaj, Woosuk Kwon, Siyuan Zhuang, Frank Sifei Luan, Gautam Mittal, Scott Shenker, et al. 2023. {SkyPilot}: An Intercloud Broker for Sky Computing. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 437–455.

[105] Murtaza Zafer, Yang Song, and Kang-Won Lee. 2012. Optimal bids for spot vms in a cloud for deadline constrained jobs. In *2012 IEEE Fifth International Conference on Cloud Computing*. IEEE, 75–82.

[106] Haidong Zhao, Zakaria Benomar, Tobias Pfandzelter, and Nikolaos Georgantas. 2022. Supporting Multi-Cloud in Serverless Computing. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 285–290.

[107] Liang Zheng, Carlee Joe-Wong, Chee Wei Tan, Mung Chiang, and Xinyu Wang. 2015. How to bid the cloud. *ACM SIGCOMM Computer Communication Review* 45, 4 (2015), 71–84.