IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 June 2023, accepted 13 July 2023, date of publication 2 August 2023, date of current version 22 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301177

RESEARCH ARTICLE

VESTEC: Visual Exploration and Sampling Toolkit for Extreme Computing

Urgent Decision Making Meets HPC: Experiences and Future Challenges

MARKUS FLATKEN[1], ARTUR PODOBAS[2], RICCARDO FELLEGARA[1], ACHIM BASERMANN[3], JOHANNES HOLKE[3], DAVID KNAPP[3], MAX KONTAK[3], CHRISTIAN KRULLIKOWSKI[4], MICHAEL NOLDE[4], NICK BROWN[5], RUPERT NASH[5], GORDON GIBB[5], EVGENIJ BELIKOV[5], STEVEN W. D. CHIEN[2], STEFANO MARKIDIS[2], PIERRE GUILLOU[6], JULIEN TIERNY[6], JULES VIDAL[6], CHARLES GUEUNET[7], JOHANNES GÜNTHER[8], MIROSLAW PAWLOWSKI[9], PIERO POLETTI[10], GIORGIO GUZZETTA[10], MATTIA MANICA[10], AGNESE ZARDINI[10], JEAN-PIERRE CHABOUREAU[11], MIGUEL MENDES[12], ADRIÁN CARDIL[12], SANTIAGO MONEDERO[12], JOAQUIN RAMIREZ[12], AND ANDREAS GERNDT[1]

[1]German Aerospace Center (DLR), Institute for Software Technology (SC), Software for Space Systems and Interactive Visualization, 31108 Braunschweig, Germany
[2]Royal Institute of Technology (KTH), Department of Computer Science, 114 28 Stockholm, Sweden
[3]German Aerospace Center (DLR), Institute for Software Technology (SC), High-Performance Computing, 51147 Cologne, Germany
[4]German Aerospace Center (DLR), German Remote Sensing Data Center (DFD), Geo-risks and Civil Security, 82234 Munich, Germany
[5]The University of Edinburgh, EPCC, EH8 9BT Edinburgh, U.K.
[6]Department of Computer Science, Sorbonne Université (SU), 75005 Paris, France
[7]Kitware SAS, 69100 Villeurbanne, France
[8]Intel Deutschland Gmbh, 85622 Feldkirchen, Germany
[9]Intel Technology Poland, 80-298 Gdansk, Poland
[10]Bruno Kessler Foundation (FBK), 38123 Trento, Italy
[11]LAERO, Université de Toulouse, CNRS, UT3, IRD, 31400 Toulouse, France
[12]Tecnosylva S.L.U., 24009 León, Spain

Corresponding authors: Markus Flatken (markus.flatken@dlr.de), Artur Podobas (podobas@kth.se), and Riccardo Fellegara (riccardo.fellegara@dlr.de)

This work was supported by the European Commission Grant H2020-FETHPC-2017 "VESTEC" under Grant 800904 [11].

ABSTRACT Natural disasters and epidemics are unfortunate recurring events that lead to huge societal and economic loss. Recent advances in supercomputing can facilitate simulations of such scenarios in (or even ahead of) real-time, therefore supporting the design of adequate responses by public authorities. By incorporating high-velocity data from sensors and modern high-performance computing systems, ensembles of simulations and advanced analysis enable urgent decision-makers to better monitor the disaster and to employ necessary actions (e.g., to evacuate populated areas) for mitigating these events. Unfortunately, frameworks to support such versatile and complex workflows for urgent decision-making are only rarely available and often lack in functionalities. This paper gives an overview of the VESTEC project and framework, which unifies orchestration, simulation, in-situ data analysis, and visualization of natural disasters that can be driven by external sensor data or interactive intervention by the user. We show how different components interact and work together in VESTEC and describe implementation details. To disseminate our experience three different types of disasters are evaluated: a Wildfire in La Jonquera (Spain), a Mosquito-Borne disease in two regions of Italy, and the magnetic reconnection in the Earth magnetosphere.

INDEX TERMS Scientific visualization, high-performance computing, topological data analysis, in-situ processing, interactive data processing, ensemble simulation, decision making.

The associate editor coordinating the review of this manuscript and approving it for publication was Mu-Yen Chen.

I. INTRODUCTION

Urgent decisions are made by a wide range of public bodies, officials and scientists in response to dangerous or

disastrous events. These events are mostly driven by the ongoing growth in global warming and its associated climate change is materializing in an increase of natural disaster events. Examples of such natural disasters include Wildfires (e.g., in Portugal, Croatia, and France) in 2017 [1], the (re-)occurring Hurricanes/Typhoons in the Atlantic and Pacific oceans [2], or extreme flooding such as the one that happened in Germany in 2021 [3]. Also, the progressive expansion of non-native mosquito species raises concern on the increased risks of large epidemics such as dengue, chikungunuya, and Zika [4]. Lastly, geomagnetic storms in the Earth's magnetosphere are taxing trillions of dollars [5] in repairs through disruptions in power grids, communication infrastructure, and satellite operations.

With disasters affecting the local population, infrastructure and environment, national and international authorities and organizations started to analyze the evolution of the incident to mitigate its effects. In this sense, simulation platforms are becoming a necessary tool to support strategic analysis for agencies, including operational units and incident field crews, dispatch centers, and incident command posts [6], [7]. In fact, many public and private entities such as forest public agencies, electrical utilities, insurance and forestry companies are already relying on science-based decision making systems. These enable them to assess hazard and risk and make decisions in real-time (e.g., CAL FIRE and electrical utilities are using Wildfire Analyst®).

The urgent decision-making process involves multiple stages starting with the acquisition of the latest data related to the incident. Such data ranges from static topographic datasets, like digital elevation models (DEMs), to live sensor data describing the actual incident in physical units. An example is hotspot data provided trough satellites, describing the occurrence of above-normal thermal radiation when investigating wild fires. In order to evaluate the effects of the disaster on additional entities, such as infrastructure, more datasets are required. For critical infrastructure, these might be data about buildings or powerlines allowing for a detailed impact assessment.

Having the initial datasets acquired, domain experts then trigger the disaster simulation. The challenge then is to continuously update the simulation. This process not only involves a continuous stream of new input data, but also deals with forces in the field providing crucial updates on the current situation. The urgency in decision-making, the complexity of models, the uncertainty of nature, and the amount of data to be processed and analysed, often limit the use of such simulation approaches in real-time scenarios.

High-Performance Computing (HPC) is the field of using supercomputers to solve large, computationally demanding problems. Today, HPC systems are reaching an ExaFLOP (10^{18} operations per second) of performance,[1] and thus their capabilities allow for estimating risks for different natural hazards, the use of stochastic approaches to assess risks and potential impacts at large scales and permitting the kinds of data-driven workflows required by urgent computing. Such supercomputers can facilitate the (near real-time) modeling and simulation of complex phenomena, such as simulating how wildfires or diseases spread or how the magnetic reconnection in the Earth magnetosphere evolves. Furthermore, by tightly integrating and augmenting these simulations with real-time sensor data (for example, near real-time satellite information of hotspots) located in close proximity to the disaster, we can:

- estimate how these disasters will unfold *ahead of time* by including multiple alternative timelines through ensemble simulations;
- quantify their potential economical and societal expected impact in order to develop mitigation actions such as to evacuate populated areas, reduce disease transmission via public health interventions, or alter satellite trajectories for better monitoring affected areas.

The idea of combining multiple sources of information (e.g., sensor data, big data, predictive simulations and analysis, visualization, etc.) is by itself not new [8], [9], [10], but orchestrating and supporting these complex workflows that leverage and exploit supercomputing resources remains mostly unexplored.

Thus, there is a strong need for a software framework which enables to: (i) orchestrate complex workflows on HPC resources, (ii) integrate and link sensor data acquisition with high-performance simulators, (iii) integrate and run ensemble simulations (e.g., multiple perturbed) for various domains, (iv) exploit advanced in-situ data analysis to extract key information, (v) interactively visualize the results in order to assess uncertainty, (vi) simplify data accessibility through an easy-to-use graphical frontend for decision-makers and domain experts.

In this paper, we give an overview of the VESTEC project [11], which was an interdisciplinary project driven by a consortium of both computer scientists and domain experts. Throughout the project we developed a prototype of a general-purpose framework that integrates extreme-scale computing for urgent decision-making.

As depicted in Figure 1, VESTEC is foreseen to be used in scenarios where a dangerous or critical event is detected and transmitted to a crisis management center that can immediately launch, monitor, analyze, and act on a continuously evolving high-precision forecast of the detected critical situation. The main contribution of the VESTEC system is to bring in heterogeneous data into supercomputers, such as data coming from sensor or social networks and/or statistics from the internet. This information is used to refine and steer the disaster simulation forecast. VESTEC supports several data analysis methods to distill out the most informative pieces of ensemble simulations, which can be visually explored and analyzed. In short, VESTEC aims to provide crisis management and to support operational disaster relief staff with

[1] www.top500.org

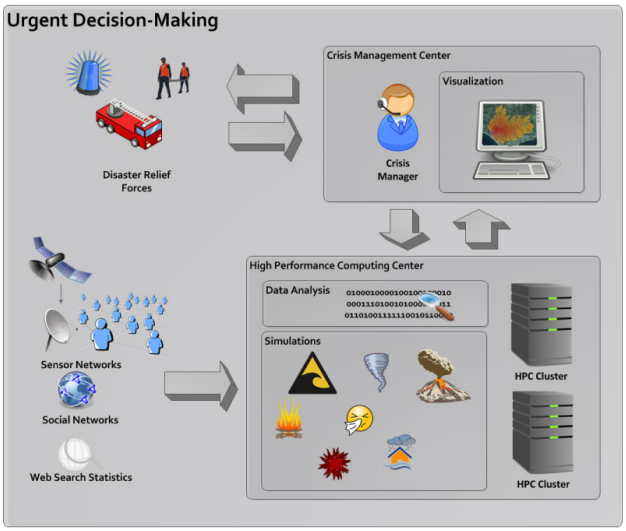


FIGURE 1. VESTEC Overview: External data sources, e.g. provided by satellites, are incorporated into extreme scale ensemble simulations executed on high-performance computing machines. Crisis management centers can guide the disaster relief forces based on the insight they got from interactive data analysis and visualization. Additionally, feedback from the forces in field should help the crisis manager to adapt and tweak the simulation scenario.

the best possible knowledge of the evolving situation. Crisis management staff can also inject domain-specific knowledge into the VESTEC system to guide the evolving simulation, which can then be used to influence the mitigation actions prompted by the crisis manager. Since the source code is released on GitHub,[2] future extensions can come directly from the community.

To demonstrate the scalability and effectiveness of the VESTEC system at supporting urgent decision-makers, we evaluate the system on three different use-cases:

- Probabilistic forest fire forecasting and monitoring;
- Mosquito-borne diseases spreading;
- Space weather forecasting.

The paper and its content are organized as follows. First, in Section II, we describe project background and related activities in urgent computing, exascale and in-situ visualization, and topological data analysis. In Section III, we introduce the VESTEC system, in which we describe the key components of the system and reveal implementation details. In Section IV, we describe and introduce the three domain-specific use cases that we integrated in VESTEC, describing their importance, their background, and how they are simulated. In Section V, we evaluate the use-cases, showing how the VESTEC system can be used to combat natural disasters or diseases. Finally, in Section VI we draw some concluding remarks and discuss directions for future work.

II. BACKGROUND

In this section, we present key background notions and activities rooted in urgent computing, exascale and in-situ visualization, and topology data analysis linked to the VESTEC system.

A. URGENT COMPUTING

Even if HPC machines have a long history of simulating disasters after the fact happened, it is less common for these to be exploited in real-time as soon as a disaster is unfolding. This requires data-integrated predictive HPC simulation with the exploitation of real-time sensor data. Historically, the SPRUCE project [12] experimented with such uses of HPC, but this was enabled via the fairly static approach of providing users with tokens that could then be redeemed to run urgent jobs manually. The urgency of their workload (e.g., whether they would only be given priority in the queue or additionally would interrupt running jobs) determined the number of tokens deduced for each run.

However, whilst such approaches as SPRUCE demonstrated promise, the approach of manually submitting urgent jobs based on tokens proved to be rather inflexible and very much a solution for the early 2000s. By contrast, modern urgent decision-makers require rich visualizations involving insights gained from the latest real-time data. Therefore a variety of workloads must run, but the exact time that they need to execute is often unpredictable (e.g., driven by the arrival of data). This dynamic behaviour is further complicated by the fact that, additionally, individual simulations can vary considerably in computational intensity, and simulation execution can be driven in real-time by emergency responders.

Then, the challenge of running urgent workloads on HPC systems is not just to run a specific predefined simulation promptly as provided by SPRUCE, but instead to enable the timely execution of many different coupled codes, driven by the unpredictable arrival of data and computational steering. Therefore, this represents a much more complex problem than the one that has been solved previously.

Put simply, modern HPC machines are not designed for such real-time workloads. There are two general reasons for this:

(i) their batch queue systems are designed for overall job throughput rather than minimising the wait and execution time associated to each individual job;
(ii) they do not operate within Service Level Agreements (SLAs) constrains required by emergency workloads.

Given the prevailing research nature of many supercomputers, it is not unusual for these machines to undergo some degree of down time which, whilst annoying for research workloads, could be catastrophic for emergency ones. While the reader might assume the solution is to install a dedicated machine for performing urgent simulations, this is often not practical. Supercomputers powerful enough to handle latest generation HPC workloads, involved in urgent simulations, are extremely expensive and, thus, these dedicated machines tend to be rather small. Consequently, most of the time such machines are idle, while during a disaster they are often not powerful enough to meet the requirements of the workloads.

[2]https://github.com/VESTEC-EU

In the short to medium term, it is possible to provide technological solutions to these issues in a manner that enables the rich flexibility required by emergency responders, which indeed the VESTEC system, as described in Section III, does. However, to properly support the processing of urgent emergency workloads on HPC machines, policy changes are required on behalf of the supercomputing sites. Such policy changes can be a significant hurdle requiring numerous approvals. Thus, we believe that the VESTEC system has an important role to play in demonstrating the benefits that can be delivered to facilitate these long-term changes.

B. EXASCALE AND IN-SITU VISUALIZATION

Producing accurate forecasts in case of an unfolding disaster, by combining real-time sensor data with numerical simulations, lead to enormous data amounts. The sheer volume of the raw data and the necessity to quickly provide insights in order to improve decision-making involves challenges for efficient data analysis and visualization. Especially in the era of exascale computing, data analysis and visualization as well as the corresponding algorithms have to be re-designed to accommodate highly parallel data processing and interactive user investigation.

In order to deal with these massive scientific datasets produced by the urgent workloads, many use case specific software solutions have been developed. Two well known and generic applications in the field of scientific visualization are ParaView [13] and VisIt [14]. Both are based on the Visualization Toolkit (VTK) [15] and exploit a client/server architecture to incorporate HPC resources for parallel processing.

While offloading these heavy processing workloads to the HPC machines greatly improved the time to solution, a main bottleneck to tackle is data I/O. In classical post-processing, the data is stored over multiple files on a distributed file system. These files are then analyzed and visualized afterwards. Just loading the data from the file system into the main memory of the machines accounts for a large portion of the total analysis time and often prohibits interactivity. An increasing performance gap between the CPU and file system exacerbates this problem and classical post-processing becomes often impractical.

To address the I/O bottleneck, a solution is to operate the analysis and visualization pipeline directly on the HPC machines while the simulation runs. This approach, known as *in-situ* processing, allows the analysis and visualizations steps to directly access the simulation data in main memory and therefore strongly benefit from the supercomputer processing power. An overview of in-situ processing techniques for large scale scientific visualization is given in [16].

In-situ analysis and visualization is enabled in ParaView by using the Catalyst library [17], [18]. Each simulation code integrates a specific adaptor that passes its internal data to VTK in a 0-copy way, i.e., by just sharing memory locations without allocating new memory. Then, the data is forwarded from the simulation model to the analysis and visualization pipelines in order to perform, for example, Topological Data Analysis (TDA) (cf. Section II-C), data reduction, or enabling an in-situ live visualization. The developed data analysis pipelines running on top of ParaView rely on the Topology ToolKit (TTK) [19], [20] in order to compute topological abstractions. The Topology ToolKit (TTK) is an open-source software library, available on GitHub,[3] for Topological Data Analysis. As soon as the data analysis step starts providing results, the ParaView (or ParaView Lite) graphical user interface is used to visually explore them.

Besides classical rasterization with OpenGL, which usually requires a dedicated GPU for rendering, ParaView additionally provides a CPU based ray-tracing engine. This engine is implemented using the Intel OSPray open-source library. Intel OSPRay [21], [22] is highly optimized for scientific visualization and supports the combined rendering of volumetric and geometric data. Since compute nodes on a supercomputer are often not equipped with dedicated graphics hardware, OSPRay utilizes the massive parallelism of CPU cores available on modern HPC systems via multi-threading and vectorization. Also, the implemented ray-tracing-based rendering algorithms, in combination with physically-based materials, enable high-fidelity photo-realistic visualizations at interactive speed.

C. TOPOLOGICAL DATA ANALYSIS

To generate insight into simulation ensembles of an unfolding disaster, efficient data analysis has to be applied. These analysis algorithms should support:

(i) the extraction of the most dominant features from all simulations in the ensemble;
(ii) the comparison of simulation runs from the overall ensemble by creating a compact representation for each simulation member. These representations will help cluster the simulations and identify outliers;
(iii) the compression of the simulation results for efficient local rendering.

Topological Data Analysis (TDA) [23] has demonstrated over the last two decades its utility to support interactive visualization tasks [24]. It robustly and efficiently captures, in a generic way, the features of interest in scalar data. These features are stored within concise, yet informative, topological data signatures. As discussed in Section III-D, these signatures are typically orders of magnitude smaller than the data itself and they can be used as a proxy to the data. Examples of successful applications of TDA include e.g. turbulent combustion [25], [26], [27], material sciences [28], [29], [30], [31], nuclear energy [32], fluid dynamics [33], medical imaging [34], chemistry [35], [36], [37] or astrophysics [38], [39] to name a few. An appealing aspect of TDA is the ease it offers for the translation of domain-specific descriptions of features into topological terms. Moreover, to distinguish noise from features, concepts from *persistent homology* [23],

[3]https://github.com/topology-tool-kit

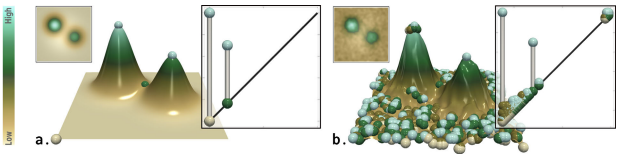


FIGURE 2. Critical points (spheres, light brown: minima, light green: maxima, other: saddles) and persistence diagrams of a clean (a) and noisy (b) 2D scalar field. From left to right: 2D data, 3D terrain visualization, persistence diagram. In both cases, the two main hills are clearly represented by salient persistence pairs in the diagrams. In the noisy diagram (b), small pairs near the diagonal correspond to noisy features in the data. ©2020 IEEE. Reprinted, with permission, from IEEE [62].

[40] provide importance measures, which are both theoretically well established and meaningful in the applications, and particularly useful for multi-scale data representations [41], [42], [43]. Among the existing abstractions, such as *contour trees* [44], [45], [46], [47], *Reeb graphs* [48], [49], [50], [51], or *Morse-Smale complexes* [52], [53], [54], [55], we focus in the following on the *persistence diagram* [40]. Its conciseness, stability [56], and expressiveness make it an appealing candidate for data summarization tasks. For instance, its applicability as a concise data descriptor has been well studied in data science [57], [58], [59], [60], [61]. In visualization, it provides visual hints about the number, data range and salience of the features of interest, which helps users visually apprehend the complexity of their data, and distinguish salient features from noise.

Given an input scalar field f (provided on a 1D, 2D or 3D triangulation or regular grid), its *persistence diagram* $\mathcal{D}(f)$ is a concise and stable topological signature which encodes the topological features of the data as a function of their salience [23]. Technically, this is a 2D point cloud (illustrated in Figure 2), where each off-diagonal point denotes a topological structure. Its X coordinate corresponds to the f value of the creation of the topological structure, while its Y coordinate corresponds to the f value of its destruction, as one continuously sweep the data by increasing f values. Its vertical distance to the diagonal denotes its salience and is called topological persistence. In the examples of Figure 2, two prominent structures (i.e. the two hills of the terrain) correspond to the two points standing away from the diagonal, while low amplitude noise (b) corresponds to points in the vicinity of the diagonal.

Generally, persistence diagrams can be computed by matrix reduction [23]. For special cases, such as saddle-extremum diagrams (which are particularly relevant in our applications), they can be computed very efficiently from merge trees [46]. Their implementation is publicly available in TTK [19], [20].

When dealing with time-varying data and ensembles, we have to use a methodology for comparing them, or technically, measuring their topological distance. The distance between two diagrams $\mathcal{D}(f)$ and $\mathcal{D}(g)$ can be evaluated with an established metric, called the L_2-Wasserstein distance [23], [63], [64], [65], noted $W_2\big(\mathcal{D}(f), \mathcal{D}(g)\big)$. It can be computed by solving an optimal assignment problem, a notoriously computationally expensive task, for which exact [66] and approximate [67], [68] algorithms exist.

Given this distance metric, datasets can then be compared by comparing their persistence diagrams. This paves the way for several advanced statistical analysis of ensemble data. In particular, given a set $\mathcal{S}_\mathcal{D} = \{\mathcal{D}(f_1), \ldots, \mathcal{D}(f_N)\}$ of persistence diagrams (typically corresponding to the diagrams of an ensemble of N scalar fields), let $F(\mathcal{D})$ be the Fréchet energy of the set, under the metric W_2:

$$F(\mathcal{D}, \alpha) = \sum_{\mathcal{D}(f_i)\in\mathcal{S}_\mathcal{D}} W_2\big(\mathcal{D}, \mathcal{D}(f_i)\big)^2. \tag{1}$$

Then the diagram $\mathcal{D}^* \in \mathbb{D}$ (where $\mathbb{D}$ is the space of persistence diagrams) which minimizes $F(\mathcal{D})$ is called the Wasserstein barycenter [65] of the set $\mathcal{S}_\mathcal{D}$ (or its Fréchet mean under the metric W_2). From a practical point of view, this is a persistence diagram which is well representative of the set $\mathcal{S}_\mathcal{D}$ (i.e. of the ensemble). Then, this diagram barycenter can act as a visual summary of an entire ensemble, but it can also be used in more advanced analysis tasks. In particular, it can be used within the popular k-means algorithm, to derive a topological clustering of the ensemble (in the space of persistence diagrams) which is faithful to the topological features found in the ensemble.

III. THE VESTEC APPROACH

Figure 3 illustrates the overall architecture of the VESTEC system, where a wide range of urgent decision-makers interact with the system via their familiar, existing, domain-specific tools. These integrate with the core VESTEC system (the blue component of Figure 3), also known as the marshalling and control system. This marshalling and control system can be thought of as middleware which controls the supercomputers and simulation runs. The interaction between client applications in yellow and the VESTEC system in blue occurs via two routes; a RESTful API and the External Data Interface (EDI). The RESTful API is a predefined set of services for undertaking activities common to many urgent workloads including the management of incidents, retrieval of data, and tracking of performance statistics. Additionally, each type of urgent situation often provides bespoke data to the system for the underlying simulations to action. This is where the EDI comes in since it can operate in both pull and push modes. The pull mode is designed for data that will be provided by specific sources, such as satellites or social networks. Conversely, the push mode is designed for the situation where a data source explicitly sends data to the VESTEC system, for example, by real-world activities occurring in the urgent environment. In both approaches, the EDI is driven by rules that match the source of the data with information where it should be posted to for ingestion by the system. Moreover, the push ingestion mechanism effectively provides a flexible API that tools can use in a domain-specific manner, for instance pushing in run-specific configuration that the RESTful API has not envisaged.

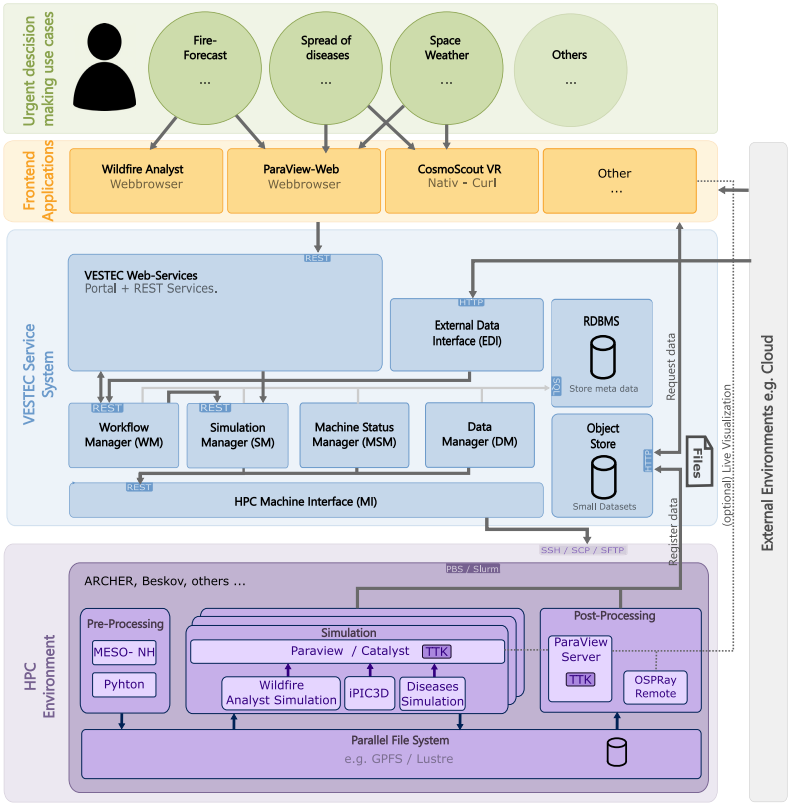


FIGURE 3. VESTEC Architecture: End-users (green) do interact with front-end applications (yellow) in order to steer and analyse their use cases. The VESTEC system (blue) abstracts and hides the complexity of workflow execution, job submission, data transfer, and job monitoring by providing flexible services via a RESTful API. Simulations, in-situ data processing and analysis is executed on remote HPC environments (purple).

As a technical solution to address the unsuitability of current generation HPC machines to run real-time urgent workloads, as highlighted in Section II-A, the VESTEC system is designed to federate over multiple HPC machines, for instance, all the supercomputers of Europe. It then determines on a simulation-by-simulation basis where to execute the simulations. This trade-off is based on calculated metrics including predicted queue wait time, application suitability for a given machine, and overhead of any data movement. Additionally, this approach solves the reliability SLA issue highlighted in Section II-A since the VESTEC system continuously monitors the progress of submitted and running simulations. In case of a system failure, it will re-submit the job elsewhere, in a way transparent to the user.

A. VESTEC ARCHITECTURE AND WORKFLOWS

Urgent workloads typically comprise numerous repeatable steps, each of which depends upon specific conditions such as the completion of proceeding simulations or the arrival of input data. Throughout this research, we have found it is natural to describe these as workflows, which is provided by the workflow manager in blue of Figure 3. These workflows are built upon the Advanced Message Queuing Protocol (AMQP) [69] and the RabbitMQ framework [70]. RabbitMQ provides an implementation of AMQP which is used as the substrate for communicating messages between workflow stages and marshalling their activation. Each urgent workload, defined per use-case, is provided to the system as a set of annotated Python functions that make up an overarching workflow, with triggers fired that will schedule other stages to run [71]. The EDI, handling the arrival of data, integrates closely with this. The data is forwarded to the appropriate workflow stage on arrival, and the workflow stage handling this as appropriate. Using this approach, all use-case-specific functionality is provided in a structured manner. Thus, it enables different workflows to plug in and out, and enables the system to handle a variety of situations. Additionally, this approach is agnostic to the data formats. The applications triggered during the execution of the workflow are responsible for data handling.

The majority of the VESTEC system (illustrated in Figure 3) can be considered as a set of underlying services which abstract the mechanism of running urgent workloads across multiple HPC machines. When integrating new use-cases with the VESTEC system, programmers need to develop their own bespoke workflows. These workflows call lower-level managers routines via pre-defined APIs.

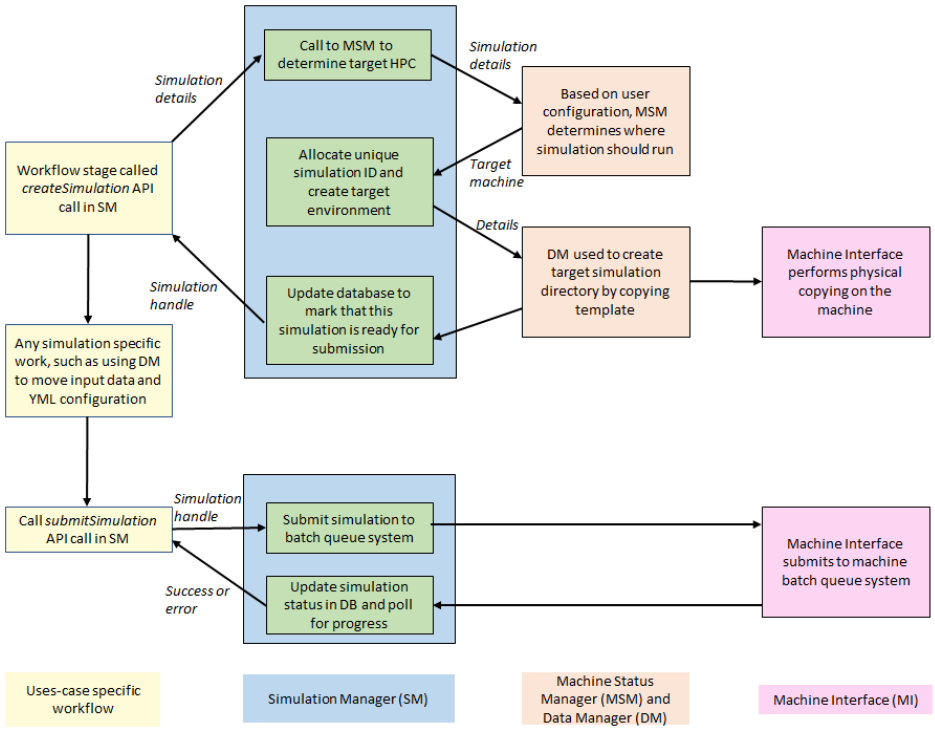


FIGURE 4. Example of interaction between a use-case workflow and the services forming the VESTEC system.

An example of this is the scheduling of simulations on the HPC machine (see Figure 4). To achieve this, the use-case-specific workflow stage will trigger the appropriate API call of the Simulation Manager (SM) with details of the job to run, input data-sets required, and optional workflow callbacks to be executed when the job reaches specific states. The Simulation Manager then calls the Machine Status Manager (MSM) that continually polls the status of connected HPC machines. MSM determines the exact machine used for execution based upon a machine learning model. This model predicts the queue wait time correlated against application suitability. Once the target machine is determined, the SM then calls the Data Manager (DM) to undertake any required data movement to set up the simulation on this specific supercomputer. The physical connection to HPC machines is handled by the Machine Interface (MI), providing a set of services to the simulation, machine status, and data managers, which are translated into appropriate commands depending upon the specifics of the machine. For instance, if an HPC machine is connected to the system via OpenSSH and is using the Simple Linux Utility for Resource Management (SLURM)[4] batch queue system, the machine interface handles this by using the OpenSSH adaptor with SLURM commands generated for gathering the status of the queue, and for submitting and cancelling jobs. All of this is abstracted from the workflow stage, with a unique identifier. This identifier can be used to retrieve further information for that job, e.g. the queue used, the number of compute nodes utilized, or sizes of memory allocated.

B. SUPERCOMPUTER SIDE ORCHESTRATION

The VESTEC system federates over multiple HPC machines and manages submitted simulations. However, at the individual machine level, there is a significant complexity that must be managed. This includes how to physically configure different simulations, how to couple simulations, and how to undertake any required data movement. Ideally, all of this should be achievable in a machine-independent manner so that the VESTEC system can remain system agnostic. One might assume that a shell script approach could be used, however, this was quickly found to be brittle and lacking generality [72].

Furthermore, there is also a trade-off between the flexibility provided by fine-grained HPC machine federation and the overhead that this entails. Therefore, the ability to couple simulation executions on an HPC machine, such that upon the completion of a simulation the next one will run automatically, is beneficial. From the VESTEC system, these coupled execution can be seen as a single atomic step, requiring only one queue submission.

To meet these requirements, we use the Common Workflow Language (CWL) [73] on the HPC machine side. CWL is a mature standard for workflow description which has gained popularity in numerous fields. Therefore, CWL provides a well-engineered reference execution tool to run

[4]https://slurm.schedmd.com/

the HPC machine-side workflows. Hence, all workloads are described in CWL. The script is then submitted to the batch queue system, with the CWL system then deciding what applications to execute and when [74]. In addition to providing a convenient approach for coupled application execution, CWL also presents a structured way of configuring the applications themselves. In this manner, skeleton configurations are provided via CWL, with run specific parameters then injected via YAML files. In this way, the core CWL configuration is machine agnostic, enabling quick and easy porting of codes between supercomputers, with machine-specific specialisation provided by a single YAML file.

A VESTEC workflow targeting a particular urgent decision-making scenario invokes the execution of multiple applications as depicted in the purple box of Figure 3. The workflow is then defined in CWL and includes applications for pre-processing, simulation models, and post-processing. For example, a CWL configuration for such a workflow first prepares the input data, then executes a numerical simulation model for performing the forecast, and finally computes key statistics in a post-precessing step.

The VESTEC system automatically determines the appropriate level of granularity for executing these stages, depending on the state and load of the HPC machines. For example, if the available resources are limited, then all stages are submitted as a single atomic step. Conversely, if there are enough resources available, logically grouped steps will be submitted and executed concurrently. This is entirely abstracted from the user and workflow.

C. EDGE/IoT SENSORS AND DATA

As described earlier in this section, data is ingested into the VESTEC system via the External Data Interface (EDI). It can operate in both a pull and push mode to best suit the data source. Endpoints in the EDI are registered by the use case. This is typically done during the initialization of workflow stages. The stage set up rules about which workflow stage should be activated when data arrives from a defined source.

As described in Section IV, the use cases of VESTEC exploit different sets of input sensor data. At the beginning of the project, we initially defined the term sensor as the measuring tool being mounted on a remote sensing apparatus. In addition to that sensor, datasets store dynamic measurements that are highly dependent on space and time. Throughout the integration of the different use cases, we concluded that the initial sensor data, e.g. obtained by satellite sensors, sometimes need to be validated and edited by the user. Therefore, we introduce the possibility to create new input data that were not acquired by the sensor itself but instead collected by forces on the field. With these feature, we consider the user itself as a source of sensor data.

D. TOPOLOGY DATA ANALYSIS FOR URGENT DECISION MAKING

As introduced in Section II-C, Topological Data Analysis (TDA) has demonstrated over the last decades its utility to support interactive analysis and visualization. In VESTEC, for each workflow, we want to exploit TDA in order to generate descriptive representations of the raw data. Since, we deal with ensemble simulations and large-scale data, we have to define efficient strategies to compute such topological representations. First, we define an in-situ computation pipeline for extracting such proxies during simulation, and then, in a post-processing step, we perform a statistical analysis.

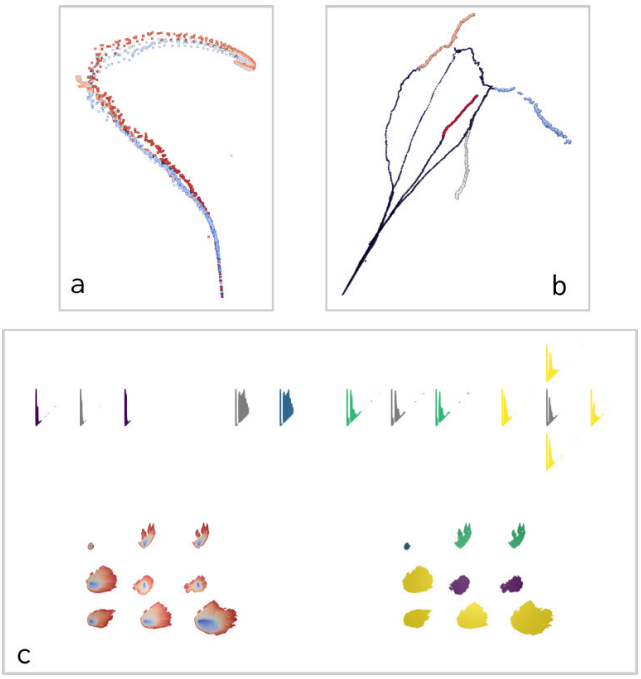


FIGURE 5. Statistical analysis of persistence diagrams on example data sets from the three VESTEC use cases (a: Embedding of persistence diagrams as a 3D point cloud through MDS for the Mosquito-Borne Diseases use case, b: Diagram embedding with additional topological clustering on the Space Weather use case, c: Topological clustering on the Wildfire use case).

1) IN-SITU COMPUTATION OF TOPOLOGICAL PROXIES

In order to enable in-situ computation in VESTEC, we rely on the Topology ToolKit (TTK) [19], [20]. TTK leverages the VTK/ParaView ecosystem, that supports in-situ computation with Catalyst [18]. Typically, an analysis pipeline using TTK modules is first modeled in the form of a Python script, which can be generated manually with a text editor or through the ParaView user interface. Next, Catalyst can be configured to run the above Python script at user-defined time steps of the simulation. Note that the simulation code must implement an interface with Catalyst, to make its internal data available in the form of VTK objects. The output of the analysis pipeline can be, itself, stored to disk if desired. In VESTEC, this strategy is particularly relevant for the in-situ computation of topological proxies. It enables to compute and store these reduced data representations, without having to store the actual simulation data, which would not be feasible in an exascale context.

The persistence diagram (section II-C) is used in VESTEC as the main topological proxy for a reduced representation of each scalar field. It is chosen for being concise, multi-scale, stable and representative of the main features of the data. Additionally, it may be desirable for some applications to store selected scalar fields for key time steps at full resolution. To cope with data size, a topology-preserving compressor can also be used [75]. The persistence diagrams and the topology-preserving compressor are computed in-situ for different time steps of the simulation. The results are then written into a CINEMA database [76], which is a SQL-type database for VTK files. This database is exploited for statistical analysis of the descriptors in a post-hoc post-processing fashion.

2) STATISTICAL ANALYSIS OF PERSISTENCE DIAGRAMS

As soon as the CINEMA databases are made available, a statistical analysis step is performed to compare different simulation runs of an ensemble.

To compare scalar fields and identify similarities or differences between the results of different simulation runs, TTK supports efficient approximations [62], [68] of Wasserstein distances (section II-C) between persistence diagrams. Given the ensemble of persistence diagrams encoded in the database, a Wasserstein distance matrix is computed and used to embed the diagrams as a 3D point cloud. This is achieved by Multi-Dimensional Scaling (MDS) [77], a standard procedure for dimension reduction in data science. In short, MDS consists in generating a point cloud (in our case in 3 dimensions, where each point represents a persistence diagram), such that pairwise Euclidean distances in 3D are optimized to coincide (as much as possible) with the pairwise distances provided by the Wasserstein distance matrix. For an ensemble of persistence diagrams, computed at different time steps for different simulation runs, this representation effectively displays dissimilarities between simulation runs.

In the context of the VESTEC project, new methods were developed to provide summarized representations of an ensemble of persistence diagrams, specifically to compute barycenters and clusters of diagrams in a progressive and efficient way [62], [78]. Such progressive algorithms guarantee time constraints, which are important for urgent decision making. These algorithms are available in TTK and used to give insight about the trend variability in an ensemble of simulation runs. The barycenters provide visual clues about the main features, while our centroid-based topological clustering finds a classification of the data based on their topology, together with a barycenter of each cluster. In practice, such a clustering enables the identification of several trends in the ensemble (in terms of features of interest), and each trend can be represented by its barycenter diagram, which is a faithful visual summary of the diagrams of a given cluster [62], [78], [79].

Figure 5 illustrates these operations on example data sets from the three VESTEC use cases. Figure 5a shows the visualization provided by the embedding of persistence diagrams with MDS for five simulation runs of the mosquito-borne disease use case spawning one year each, which illustrates the similarity between runs (one color per run). Each point corresponds to a single persistence diagram. In particular, the five runs have a similar temporal pattern, describing a closed curve (the simulation spans a full year), with sharp U-turns located at solstices. In contrast, the same representation on four runs of the Space Weather use case with different input parameters (Figure 5b) shows how the runs exhibit distinct behaviors, past a certain point in time. The embedding is augmented with the result of the topological clustering (colors) performed on the last steps of the simulations. Figure 5c shows the result of the topological clustering on a small data set from the Wildfire use case, where the clustered diagrams (top row) match a ground truth classification (bottom right). The centroids (top row, gray diagrams) are representative of the topology inside each cluster.

E. VISUALIZATION AND WEB INTERFACE

As mentioned earlier in this section, the applications highlighted in yellow of Figure 3 include the visualization and data analysis tools that the end-users can interact with. Since each use case could have different requirements, e.g., linked to user interactions and visualization techniques for their specific data, we support and integrate tailored visualization applications to demonstrate that VESTEC supports data analysis and visualization for many application domains.

Currently, the following visualization applications are integrated in the system.

1) WILDFIRE ANALYST

The Wildfire Analyst web interface, developed by Tecnosylva for the VESTEC project, is based on the Wildfire Analyst®API and allows users to quickly estimate fire behavior probabilities during forest fire incidents. It supports them in making urgent decisions such as where to locate resources, where to construct a fireline or where and when to protect or evacuate assets or specific locations. The Wildfire Analyst web frontend application directly communicates with the VESTEC system through the dedicated Restful API. The bi-directional communication allows triggering several processes in the VESTEC system according to the predefined workflows. These processes include gathering/fetching of input data, preparing the data according to the user defined area of interest, preparing the input data to be forwarded to the simulation code, storing and moving the simulation results, and providing results back to the frontend application. The Wildfire Analyst web frontend as depicted in Figure 6 is the main application for the forest fire use case. The user directly interacts with this graphical user interface in order to configure and run simulations and to assess and analyze its results.

The user interface is equipped a Geographical Information System (GIS) component that has been specifically designed and developed for urgent decision-making purposes

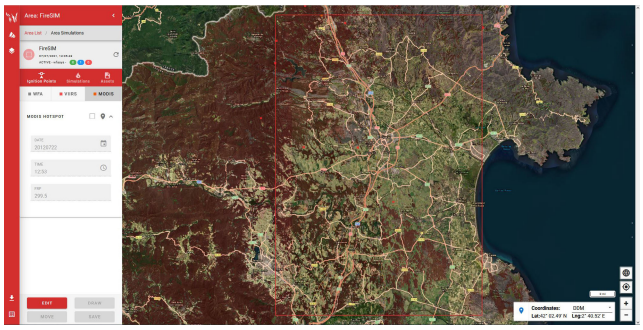


FIGURE 6. The wildfire analyst web user interface developed in VESTEC.

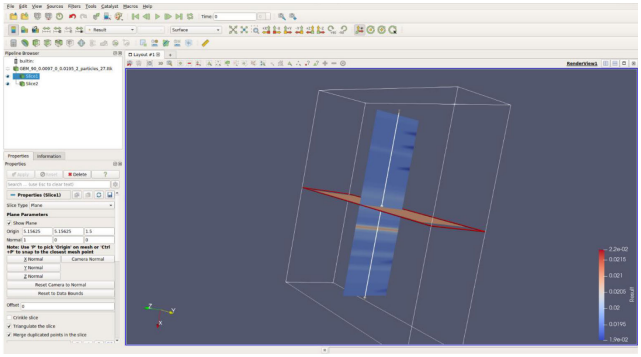


FIGURE 7. ParaView interface with an analysis pipeline showing two slices of a simulated Space Weather magnetic field.

in VESTEC to support the forest fire use case (described in Section IV). It allows users to visualize and edit hotspot data, e.g., captured by remote sensing satellites. Users can evaluate and modify their metadata, or add additional hotspots. Furthermore, the user interface allows setting forest fire simulation parameters and triggering new forest fire behavior simulations based on the hotspots or existing assets. Finally, the corresponding results of the simulation runs (static and probabilistic forest fire simulations) are visualized.

2) ParaView

Reference [13] is an open-source, multi-platform data analysis and visualization application. ParaView users can quickly build visualizations to analyze their data using qualitative and quantitative techniques (cf. Figure 7). The data exploration can be done interactively in 3D or programmatically by using ParaView's batch processing capabilities. ParaView itself is based on a client-server architecture, that enables the analysis of extremely large datasets, e.g., as produced by the space weather use case (described in Section IV). The server then exploits the distributed memory computing resources on the HPC machine. While the server now runs in parallel on the supercomputers to process datasets of petascale size, the client can be executed on commodity hardware such as a laptop. Therefore, ParaView has become a state-of-the-art tool in many national laboratories, universities, and industries, and has won several awards related to high-performance computing.

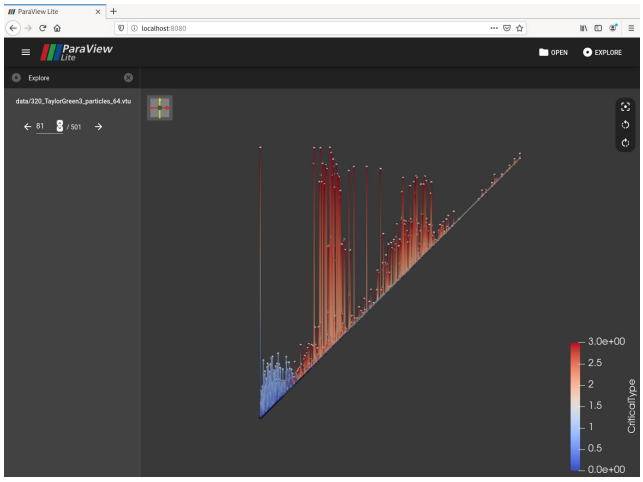


FIGURE 8. ParaView Lite is the web version used in VESTEC. The figure depicts a rendering of a persistence diagram in the web browser.

FIGURE 9. CosmoScout VR: Analysing and interacting large-scale terrain data in virtual reality environments.

Besides the classical approach of using ParaView as a desktop application, VESTEC exploits and extends ParaView Lite. ParaView Lite is a web-based visualization application based on vue.js that enables data analysis and visualization in a web browser. In practice, the frontend application connects to a running ParaView server instance on the HPC machine, which is connected to the simulation using Catalyst [17], [18]. This setup is used when doing in-situ processing. Data from the simulation is processed and rendered on the server. The frontend application, either ParaView or ParaView Lite, just receives a video stream at interactive frame rates. Within this eco-system, all functionalities available in TTK, as described in section III-D, can be exploited. Furthermore, interactive ray-tracing, based on Intel's OSPRay rendering library, can be performed.

3) CosmoScout VR

References [80], [81], as depicted in Figure 9, is a modular virtual universe developed as open-source software at the German Aerospace Center (DLR). Because of its strong focus

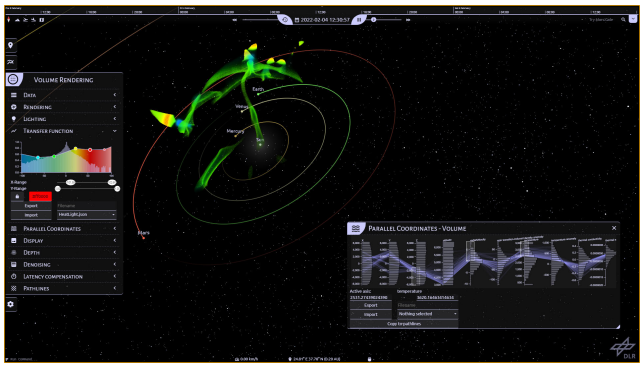


FIGURE 10. CosmoScout VR: Visualization and analysis of space weather phenomena. The parallel coordinate plot enables end-users to quickly select subdomains in the data based on multiple thresholds.

on virtual reality and user interaction, the seamless navigation allows end-users to explore massive geo-referenced datasets. This data exploration includes tasks such as measuring terrain structures, the identification of obstacles to guide relief forces, or to assess mitigation strategies for the disaster being studied.

In particular, CosmoScout VR enables 3D immersive visualization, analysis, and presentation of large-scale Earth observation and remote sensing data, and this data, in VESTEC, is then augmented with the simulation and analytic results. In addition to the visualization of 2D and 2.5D surface data, an interactive volume rendering approach based on Intel's OSPRay library [21] has been developed and integrated. This plugin enables interactive visualization of massive space weather simulation results as depicted in Figure 10.

To provide the end-users with detailed visualizations of context information, e.g., through monthly temperatures or precipitation values, the interfaces based on the Web Map Service (WMS) in CosmoScout VR have been extended to enable querying time dependent data. As described in Section III-D, it has been also evaluated and demonstrated the usefulness of the topological proxies for urgent decision making, by combining their visualization with the simulation results. Therefore, a brushing-and-linking approach was integrated. End-users can directly interact with the persistence diagrams. They can brush data in the diagram and, then, the selected data is immediately highlighted in the 3D view of CosmoScout VR. With this feature, end-users can quickly identify and select prominent features within the ensemble simulation and directly spot the areas of interest. Finally, the user interface has been extended with a graph-based editor, as depicted in Figure 14, to allow user-friendly data analysis. This HTTP and JavaScript-based user interface further enables direct communication with the VESTEC system by integrating the end-points of the RESTful API, described in section III. Therefore, end-users can not only fetch and visualize result data, but also login to the VESTEC system, define and trigger the execution of an incident or upload some input data for the simulation itself, e.g., the latest information from the field.

IV. URGENT DECISION MAKING: THE VESTEC USE-CASES

Real-time emergency response applications require the combination of highly parallel simulations with high-performance data analytics [82], [83], [84], [85], [86]. Classical examples for real-time disaster simulations are simulations and predictions of earthquakes, floods / tsunamis and typhoons, wildfires, diseases, critical traffic simulations, and simulations of evacuation scenarios. In the following, the VESTEC use-cases and their simulators are described: mosquito-borne diseases, magnetic reconnection in space weather, and probabilistic forest fire forecast and monitoring.

A. MOSQUITO-BORNE DISEASES

The dynamics of mosquito abundance over time represent a crucial ingredient to assess the risk of vector-borne disease outbreaks and their potential spread in human populations. The main goal of the Mosquito-Borne Diseases (MBD) use-case is to integrate innovative epidemiological models with high-performance computing approaches, data analytics processes, and visualization tools to support urgent decision-making during future epidemic threats. Such goal was motivated by the progressive expansion of tropical diseases transmitted through mosquito bites to temperate climate areas. Globalization and climate changes have led different Aedes species to expand their habitat to temperate regions. This phenomenon is raising significant concern in Europe since there could be an increasing risk of experiencing large epidemics linked to diseases that have been primarily characterized by sporadic and short transmission chains. The risk associated with mosquito-borne diseases largely depends on the abundance of competent vectors for the transmission of the infection. The estimate of the potential vector abundance is a crucial issue for the public health decision-makers, as the CDC Epidemic Prediction Initiative well documents [87]. Existing computational models cannot estimate the absolute abundance of different mosquitos' species at high resolution over large spatial scales, nor update estimates in nearly real-time to reflect changes in the meteorological or epidemiological conditions. A novel modeling approach extending on [88] was therefore developed, calibrated as informed by a large set of entomological data collected in the last decades in Italy and the US, and integrated into the VESTEC system.

1) SIMULATION

The modeling approach extends [88] and it is based on two simple ideas. The first one is that the overall abundance of the vector is driven by a variety of socio-demographic and eco-climatic factors, such as gross domestic product (GDP), human density, temperature, and precipitation records. The second idea is that an increase in the abundance of adult mosquitoes occurs because of persisting favorable temperature conditions over a certain period. Specifically, we assume that the relative abundance of female adult mosquitoes at any

given day d can be approximated with a logistic function of the average temperature observed over a certain time window preceding that day and a set of static socio-demographic and eco-climatic measures associated with the geographical area considered. Leveraging on available estimates on the capture rate associated with different Aedes species and mosquito traps, the developed model can rapidly produce estimates on the absolute abundance of the vector population for each day of the year with a spatial resolution of 250m x 250m, and compute the transmission potential associated with dengue, chikungunya and Zika across any location of interest in Europe. Model parameters were calibrated separately for Ae. aegypti and Ae. albopictus through a Markov chain Monte Carlo approach applied to the Negative Binomial likelihood of observing the number of captured female adult mosquitoes over time across different years and geographical locations. The Mosquito-Borne Diseases (MBD) simulator is sufficiently flexible to be integrated with any HPC infrastructure and to be coupled with sensor data or weather simulation forecasts to provide real-time updates on the current outbreak risks.

The production of a massive variety of scenarios accounting for model uncertainties on a large spatial scale requires proper management of processing power and memory to provide timely model estimates to the end-user. Its implementation within the VESTEC framework, therefore, includes the management of simulation ensembles, progressively building up statistically accurate pictures of emerging time-critical phenomena, the integration of suitable data compression, the extraction of topological features characterizing the output data, and the appropriate visualization of results.

B. SPACE WEATHER FORECASTING

In the context of space weather simulations, our goal is to perform ensemble simulations of magnetic reconnection [89], [90] under several different space weather conditions (e.g. different solar wind velocity, and magnetic field) to enable data analysis and uncertainty quantification studies. Magnetic reconnection is an important physical phenomenon regulating the exchange of energy and momentum between the solar wind and Earth's magnetosphere [91], [92], [93]. In essence, magnetic reconnection converts the magnetic field stored in Earth magnetosphere into kinetic energy of plasma jets. Among other effects, magnetic reconnection jets are responsible for generating aurora on the Earth. The main scientific question we address with VESTEC is how the magnetic reconnection dynamics varies by changing the space weather conditions. To address this question, we spawn and monitor multiple iPIC3D simulations. iPIC3D [94] is a widely-used, massively parallel Particle-in-Cell (PIC) code, used for magnetic reconnection and magnetospheric modeling. For demonstration purposes, we use the base configuration from the GEM Reconnection Challenge [95]. The GEM challenge is a community-defined benchmark to understand which simulation model accurately captures magnetic reconnection dynamics and in particular the rate magnetic reconnection occurs. The GEM challenge simulation configuration mimics the configuration of Earth's magnetotail and can be applied for studying magnetic reconnection under different conditions varying the intensity of a background magnetic field to mimic the dayside magnetopause reconnection or the background densities of plasma populations.

For the ensemble simulations, we use different parameters characterizing the solar wind conditions, e.g., density and magnetic field convected by the solar wind and perform 3D iPIC3D simulations. For simplicity, we focus on characterizing the magnetic reconnection dynamics by varying the intensity of a background magnetic (called magnetic guide field) along the direction perpendicular to plane where the reconnection occurs (in our simulation, the reconnection develops on the $x - y$ plane so the background guide field is indicated with B_{0z}). Therefore, several pre-processed input files with different background magnetic fields are exploited. Each simulation of the ensemble provides the characteristic distribution functions to be compared directly with the observed distribution functions from the NASA Multiscale mission[5] [96] and particle-trajectories for further post-processing and data analysis [97].

1) SIMULATION

As said earlier in this section, to simulate Space Weather phenomena we use the Particle-In-Cell method (PIC) [94], [98]. The PIC method focuses on simulating the trajectories of plasma particles, such as electrons and protons. The method itself calculates the forces between particles (e.g., Coulomb and Lorentz forces) using a mean-field grid-based method. We calculate the fields on each grid point by applying and solving Maxwell's equations, where we are given the current density and charge on each point in the grid. We redirect the interested reader to excellent books on the topic of computational plasma physics [99], [100] for a more detailed explanation of the method. Implicit PIC methods, such as the one iPIC3D uses, allows for simulations with relatively large time step and grid spacing, when compared to explicit methods.

Algorithmically, the PIC method is composed of four distinct phases. The first phase initializes the particle positions, velocities, and electric/magnetic fields and is followed by the remaining three phases: Particle Mover/Pusher, Particle-to-Grid Interpolation (also known as Moment Calculation), and the Field Solver.

a: THE PARTICLE MOVER

The Particle Mover (or Pusher) is responsible for solving the equation associated with the motion of each simulated particle based on its position $\mathbf{x}_p$ and its velocity $\mathbf{v}_p$. We use an implicit in-time discretization scheme for the particle equation of motions, as measured in CGS units, and couple

[5] Available on the MMS Science Data Center (SDC) at https://lasp.colorado.edu/mms/sdc/public/

that with a predictor-corrector scheme to calculate average velocities according to $\bar{\mathbf{v}}_p = (\mathbf{v}_p^n + \mathbf{v}_p^{n+1})/2$ during the time-step Δt with n indicating the time level:

$$\tilde{\mathbf{v}}_p = \mathbf{v}_p^n + \frac{q\Delta t}{2m}\bar{\mathbf{E}}_p \tag{2}$$

$$\bar{\mathbf{v}}_p = \frac{\tilde{\mathbf{v}}_p + \frac{q\Delta t}{2mc}(\tilde{\mathbf{v}}_p \times \bar{\mathbf{B}}_p + \frac{q\Delta t}{2mc}(\tilde{\mathbf{v}}_p \cdot \bar{\mathbf{B}}_p)\bar{\mathbf{B}}_p)}{(1 + \frac{q^2\Delta t^2}{4m^2c^2}\bar{B}_p^2)}, \tag{3}$$

where p is the index of the calculated particle, q and m are the particle charge and mass, respectively, and c is the familiar speed of light in vacuum media.

The number of iterations we use to compute $\bar{\mathbf{v}}_p$ is either set by a pre-described error tolerance (configured by an expert) or manually fixed to a small number of iterations. In our implementation, we use three different iterations for the electron and proton particles. To compute $\bar{\mathbf{v}}_p$, we depend on both the electric and magnetic field for a particular particle position, $\mathbf{E}_p$ and $\mathbf{B}_p$, respectively. However, the values of $\mathbf{E}_p$ and $\mathbf{B}_p$ are only available and defined at the grid points in the PIC method, and must thus be computed using an interpolated (or weighted) function $W(x_g - x_p)$ that we define as:

$$W(x_g - x_p) = \begin{cases} 1 - |x_g - x_p|/\Delta x & \text{if} \quad |x_g - x_p| < \Delta x \\ 0 \quad \text{otherwise.} \end{cases} \tag{4}$$

Here we use a linear interpolation function ($W(x_g - x_p)$) albeit higher-order function can be used. Through interpolation, we can easily calculate both the electric and magnetic field at a particular particle position of interest from the values on the grid-point g:

$$E_p = \sum_g^{N_g} E_g W(x_g - x_p) \qquad B_p = \sum_g^{N_g} B_g W(x_g - x_p). \tag{5}$$

Knowing the average particle velocities, we now can update each particle position (and its velocity) according to:

$$\begin{cases} \mathbf{v}_p^{n+1} = 2\bar{\mathbf{v}}_p - \mathbf{v}_p^n \\ \mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \bar{\mathbf{v}}_p \Delta t. \end{cases} \tag{6}$$

For a more detailed and comprehensive mathematical derivation of the discretized equation, we refer the interested reader to [98], [101], and [102].

We would like to point out and stress that the particle mover/pusher consumes the majority of the computational time when simulating Space Weather phenomena, and therefore, is an excellent candidate for performance optimizations. For example, it is not uncommon for the particle mover/pusher to claim between 68%-73% in typical Space Weather simulations [103] (subject, of course, the studied problem at hand). Hence, in the VESTEC project, we have not only used a CPU-based particle solver but also developed a prototype GPU-based solver named sputniPIC [104] to enable performance increases on heterogeneous supercomputers.

b: PARTICLE-TO-GRID INTERPOLATION

After the particle mover/pusher phase is complete, we start the particle-to-grid interpolation phase, where we calculate any quantities that are the source (or input) for our field solver. In our implicit PIC method, these quantities include the current density ($\mathbf{J_g}$), the charge density (ρ_g), and the pressure density tensor (P_g). All of these quantities are defined on the grid points and are all calculated based on the particles' positions and their velocities. As with the calculation of the magnetic-/electric-fields in the particle mover/pusher phase, we continue to use the interpolation functions $W(x_g - x_p)$ to determine $\rho_g, \mathbf{J}_g, P_g$ at the grid point g:

$$\{\rho, \mathbf{J}, P\}_g = \sum_p^{N_p} q\{1, \mathbf{v}_p, \mathbf{v}_p \otimes \mathbf{v}_p\} W(x_g - x_p). \tag{7}$$

This phase is the second-most (after the particle mover/pusher) computationally demanding part of our PIC method, and in typical scenarios of magnetic reconnection, it can require up to 25% of the entire computational cycle [103]. As with the particle mover, the particle-to-grid interpolation is an excellent candidate for performance optimization, and in the VESTEC system, it has been extended in the sputniPIC prototype to support GPU-based systems.

c: THE FIELD SOLVER

The computational cycle in our implicit PIC method ends with the field solver, which computes the solution of the discretized Maxwell's equations on the grid. In essence, this phase takes the previously computed quantities ($\rho_g, \mathbf{J}_g, P_g$) and computes E_g and B_g. We compute said quantities by solving a linear system of equations that arise when we discretize Maxwell's equations implicitly in time using a generalized minimal residual (GMRes) [105] linear solver. Aside from GMRes, we also solve a discretized Poisson equation using the conjugate gradient (CG) method at each computational cycle in the simulation in order to ensure that the continuity equation remains satisfied [101]. Performing this additional step is often called divergence cleaning. Unlike the two previous phases, which consume a large fraction of the computational cycle, this phase requires a mere 6% of the entire computational cycle. For this reason, we can afford to leave the GMRes solver on the general-purpose host processor.

C. PROBABILISTIC FOREST FIRE FORECAST AND MONITORING

In case of a wildfire emergency, first responders require accurate prediction of:

- the fire spread and behavior,
- potential impact on communities and assets,
- associated risk indices to optimize different suppression alternative.

In this sense, Wildfire Analyst®applies research and technology together to improve the implementation of fire models

in real-time, situational awareness and decision-making in operational environments. The platform has a user-friendly software interface that includes real-time data integration (e.g. fire progression with hotspots), weather forecast services, high-resolution thematic cartography, improved fire modeling capabilities and viewers to interpret the outputs. Wildfire Analyst is the forest fire simulation software that has been integrated in the VESTEC system to address the specific emerging use case of wildfire monitoring and forecasting. The implemented code aims to improve a running wildfire simulation using a guided Monte Carlo approach. The forest fire propagation forecasting is based on the well-known quasi-empirical model proposed by Rothermel [7] and Albini [106]. The implemented model exploits a heat balance equation. It additionally requires certain parameters to be tuned based on wind tunnel experiments and well documented historical fires. This model has been used for decades as one of the main wildfire simulation tools in the U.S.A. and Europe and forms the basis of the currently most advanced forest fire simulators. In VESTEC, the used model takes the weather simulation outputs from the MESO-NH code [107] as an input for running simulations with weather variances.

1) SIMULATION

MESO-NH [107] is a limited-area atmospheric model that applies to a wide range of resolutions, from synoptic to turbulent scales. Implemented in Fortran 90 and MPI, its large parallel computing capability has been tested up to 130 000 cores to reach a performance of 4 Teraflop/s [108] and used for research purposes with 16 000 cores and more than 1 billion gridpoints [109]. The initial and boundary conditions provided to MESO-NH are given by the U.S. National Weather Service Global Forecast System (GFS). The grid configuration has a horizontal spacing of 2 km and a vertical spacing stretching from 12 m near the surface to 600 m at the top of the model. The MESO-NH outputs provide hourly records on temperature, relative humidity, wind speed and direction at 6 m height. This information is then used as input by Wildfire Analyst.

The Wildfire Analyst simulation model makes use of the main fire behaviour variables such as vegetation type, wind speed and orientation, terrain slope, vegetation moisture content, canopy cover, and height of the overstory, among other parameters. The simulation model can also take into account other phenomena such as crown fire and fire spotting. The forecasted time evolution of the forest fire progression is based on a minimum travel time (MTT) algorithm [110]. The simulation uses a non-overlapping, hybrid, OpenMP - MPI parallel implementation, making use of as many threads per rank as stated in the configuration file. This simulation code simulates probabilistic as well as non-probabilistic results of forest fire propagation. The non-probabilistic outputs include the estimation of the initial non-perturbed fire simulation. Given a set of weather inputs obtained from the MESO-NH weather model and fire ignition location(s) (i.e., hotspots)

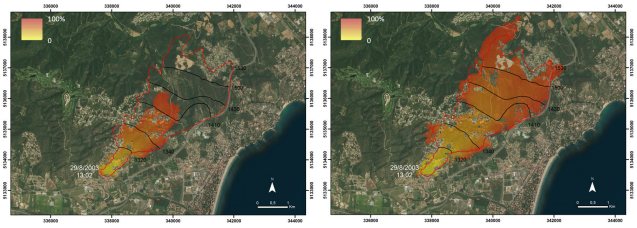


FIGURE 11. Fire time of arrival of free simulated fire growth (left) and calibrated simulated fire growth (right) considering the adjustment factors obtained through location and time control points. The fire duration of this fire example was 2.5h.

from satellite sensors or from the user itself, a first simulation is carried out representing the initial non-perturbed fire simulation. From this initial simulation, further hotspots on the terrain will be used to calibrate the rate of spread (ROS) of the simulation and update the simulation output, trying to fit this information into the simulation results. The capability of being able to adjust fire simulations is crucial in the simulation of fire behaviour. Due to inherent model inaccuracies, lack of model applicability, or erroneous input data, many times wildfire predictions become significantly inaccurate and not reliable. Data-driven techniques aim to circumvent this problem by using observed fire front data to tune or calibrate simulations to the observed fire patterns. This approach is very promising and it can be powered by observations from satellite sensors, unmanned aerial vehicles (UAVs), and GPS locations of suppression resources. A common way to reduce inaccuracy of a simulation model is to adjust the simulated fire growth to the observed fire progression by tuning the ROS adjustment factors [111]. These adjustment factors are a set of fuel-related constants ''Adjfuel'' used to modify fire's rate of spread (ROS) in a simulation in the following way:

$$ROSfinal = Adjfuel * ROSmodel \tag{8}$$

These factors are well known by fire practitioners and researchers since they provide a direct way to calibrate fire simulations for observing fire spread. Nevertheless, manually determining these factors is a hard and time-consuming task that requires patience and a good set of trial and error attempts to be completed. During a fire simulation, calibration data is given through a set of control points where the arrival time of the real fire is known, these are obtained from satellite sensors (such as VIIRS and MODIS) or directly introduced by the user through the user interface. Every time new hotspots become available, these are used by the simulation code to calibrate the performed simulations. The adjustment method is based on determining the best ROS adjustment factors using a least-squares approach to minimize the error between the simulated fire growth and the real fire [112]. An example of an adjustment of the simulation of the Castell D'Aro fire [111] is depicted in Figure 11.

The left picture shows the output of an initial non-perturbed forest fire simulation without any adjustment. The right picture shows the result of the simulation after performing the

adjustment through the use of control points. These indicate to the simulator where the fire front was located at which time and is used by the simulator to adjust the rate of spread of the fire. Then another simulation with similar conditions is executed, providing more realistic fire propagation results. Regarding probabilistic outputs, the Wildfire Analyst®simulation code creates an ensemble of semi Monte Carlo simulations with slightly different weather and vegetation moisture. These varying input conditions are used to obtain a probabilistic analysis of the fire behaviour. It also makes use of the provided control points to make a probabilistic search to find the best fit simulation among the set of simulations according to the provided input variables. This provides decision-makers with a more reliable source of data that already takes into account input or modelling inaccuracies. The main graphical outputs provided are the fire burning probability and the fire front location probability at each spatial cell of the spatial domain at a given moment in time. These outputs are composed of three preliminary main layers that are used by the code to generate the previously mentioned final outputs. These preliminary layers are:

- *rasNfire*: representing the number of times a fire is estimated to reach a given spatial cell.
- *rasMean*: representing the average time at which fires reached a given spatial cell.
- *rasVariance*: representing the time variance at which fires reached a given spatial cell.

These three base layers provide a Gaussian distribution characterization of the times at which fires reached a given spatial cell and therefore allow making use of normal distributions to compute the fire burning probability. Another output calculated by the code is the estimation of exposure fire sheds. The major concern for fire incident commanders at the time a fire starts is usually how to protect vulnerable areas such as wildland urban interfaces (WUIs), protected natural areas, or critical infrastructures. A fire behavior expert may suggest an evacuation or shelter-in-place decision based on predicted fire weather conditions, observed fire behavior, fire suppression capability, access to escape routes, and characteristics of the population or community. For this reason, having a dynamic fire arrival time for those areas to be protected is crucial to support the decision-making of the fire analyst. These exposure sheds are estimated using a simulation method in inverse mode (PIM). This allows quantifying the impacts of weather uncertainty on fire behaviour and, subsequently, the time of a wildfire reaching an asset to be protected from each spatial cell of the landscape. The PIM method is based on a Monte Carlo approach. Fire simulations in inverse mode are based on a fire inverse travel time mode technique [113], [114] that allows computing x-hour firesheds around defined exposure cells (i.e., areas to be protected such as buildings, infrastructures, nuclear power plants, etc). This fire shed output represents the time that a standard fire simulation (i.e., a fire starting from an ignition source) starting at any spatial cell in the landscape would take to reach the user-defined exposed cells. When calculating the firesheds, the source code takes the defined variations into account (e.g. weather and vegetation moisture conditions) and provides the best fit simulation result from the set of simulations calculated.

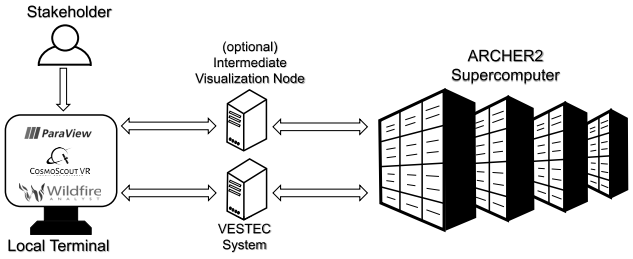


FIGURE 12. Abstract view over the use-case execution, showing key steps and components.

V. RESULTS

In this section we evaluate the VESTEC system by executing the use cases integrated in it. The list of datasets used by each use case is given in Table 1.

A. EXPERIMENTAL PLATFORM

All use-cases were executed on the ARCHER2 supercomputer provided by the University of Edinburgh (EPCC). The ARCHER2 supercomputer has a total of 5860 compute nodes, where each node contains two AMD EPYC 7742 64-core processors running at 2.25 GHz and is equipped with 256 GiB of main memory. The compute nodes are interconnected using HPE Cray Slingshot.

Additionally, the Space Weather use case has also been demonstrated on a workstation with an eight-core Xeon E5-2609v2 processor running at 2.5GHz. The workstation is equipped with a total of 72GB DRAM. The storage consists of two 4TB HDD (WDC WD4000F9YZ / non-RAID) and a 250GB SSD (Samsung 850 EVO).

In the remaining of this section, each case study is organized into four subsections: `methodology` describes the overall workflow of the use-case; `execution` qualitatively describes the use-case and how it was carried out; `performance` presents aspects related to computer performance of the use-case; and finally `relevance` discusses the importance and applicability of the use-case to key stakeholders and decision-makers.

The general flow (specific use-case descriptions are provided in the following sections) of triggering an incident (i.e., running a use-case) is shown in Figure 12 and is organized into four steps: *(i)* the user (a stakeholder) uses an intuitive frontend interface where they enter credentials and select the type of incident along with required other parameters (e.g., data sources, areas of interest, etc.), *(ii)* when the incident is started, the VESTEC system orchestrates the execution and connects the local frontend to the high-performance computing resources and launch ensembles of simulation associated with the incident, *(iii)* the supercomputer (in our cases,

TABLE 1. Sensor datasets used by the three applications in VESTEC.

Use case	Data source	Dataset
Wild fires	MESO-NH	Temperature
		Relative humidity
		Wind speed
		Wind direction
	Satellite sensors	MODIS hotspots
		VIIRS hotspots
		MSG-Seviri hotspots
	User	Custom hotspots
Mosquito-borne diseases	WorldClim [115]	Weather stations
		MODIS Land Surface Temperatures
Space weather	Satellite sensors	Fast Plasma Investigation (FPI) Ion distribution function
		FPI Electron distribution function

ARCHER2) then executes all simulations, and either stores the result locally or in case of in-situ visualization streams the results to the intermediate visualization node as the simulation unfolds, and finally *(iv)* the VESTEC system, when the simulations provide some outputs or is completed, transfers the result from the supercomputer back to the stakeholder for inspection, further analysis, or refinement/re-execution of the workflow.

B. CASE STUDY 1: URGENT DECISION MAKING THROUGH APPROPRIATE VISUALIZATION OF MOSQUITO-BORNE DISEASES OUTPUTS

The following section presents the result obtained while executing the Mosquito-Borne Diseases (MBD) use-case using the VESTEC system.

1) METHODOLOGY

The overall methodology involves the use of CosmoScout VR as the main graphical user interface to access the mosquito-borne diseases workflow and visualize the results. The ARCHER2 supercomputer system was used for performing the simulations. There are roughly four steps that are performed during the VESTEC mosquito-borne diseases use case: *(i)* a user logs in the VESTEC system via CosmoScout VR, *(ii)* the user defines the required simulation parameters (area of interest, mosquito species, disease, number of members in the ensemble simulation), and starts the workflow, *(iii)* simulations are carried out on the supercomputer, including the computation of topological proxies via TTK, and *(iv)* the result data is streamed to CosmoScout VR for interactive visualization and analysis.

2) EXECUTION

The overall goals of the mosquito-borne diseases (MBD) use-case are: *(i)* to demonstrate the integration and usage of VESTEC components to provide robust indications about epidemiological risks associated with MBD, covering a large area (e.g., continental scale) and with a spatial resolution comparable to the scale of disease spread (i.e., approximately 250m), *(ii)* to show how supercomputers can help decision-makers make fast and effective decisions regarding threats associated with MBD, and *(iii)* to concretely show

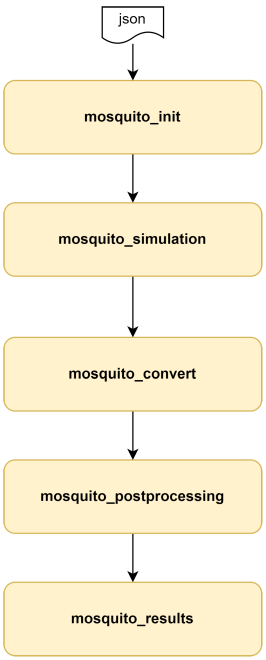


FIGURE 13. Execution workflow for the Mosquito-Borne Diseases use case.

examples of model estimates on areas with known past circulation of MBD, using ensemble simulations by leveraging the VESTEC approach.

The incident described here is based on the Lazio region in Italy, where the largest MBD oubreak caused by Aedes species in the history of continental Europe has occurred. We therefore used the model described in Section IV by selecting Aedes albopictus as the vector species, chikungunya as the disease of interest, and the the Lazio region as the area to be analyzed.

As shown in Figure 13, in *mosquito_init* step the incident is initialized with a json file containing all the user-defined input parameters, the VESTEC system orchestrates several simulations (*mosquito_simulation* step) at the supercomputer, and the user has to wait until first results are available and streamed from the VESTEC system to CosmoScout VR. The user gets a feedback about the progress of simulations directly from the user interface. Initially, the maps,

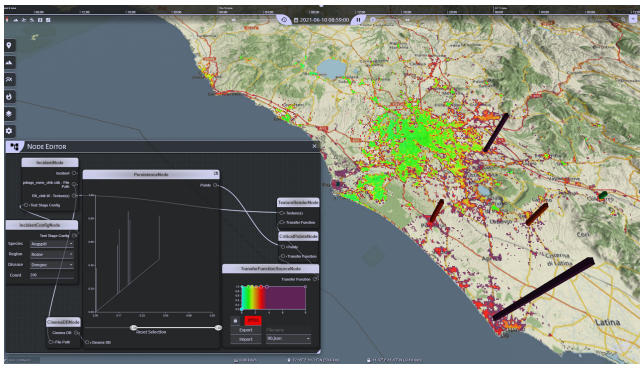


FIGURE 14. Analysis and visualization of the MBD simulation results augmented with topological information represented in the persistence diagram. Vertical bars do represent areas with highest risk of an outbreak.

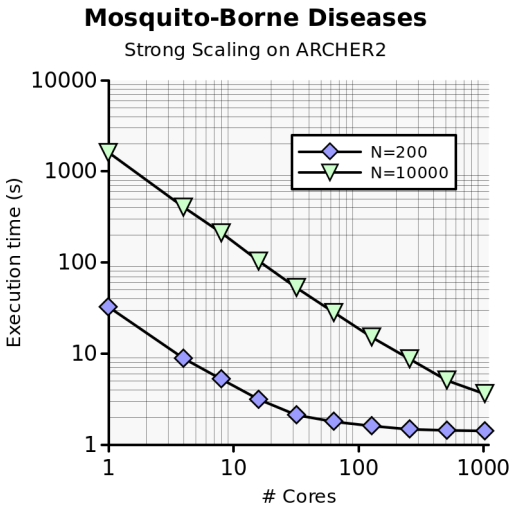


FIGURE 15. The strong-scalability performance of the mosquito-borne disease use-case, where we see that the use-case can scale up-to 1024 processing cores, demonstrating that urgent decision makers can leverage the large compute capacities of modern supercomputers to have a better basis onto-which to combat the spread of diseases.

covering the selected area and used as input, are extracted from large datasets. These maps are used to perform the simulations and include meteorological data, the population density and the gross domestic product (GDP). Thereafter, in the *mosquito_convert* step, the time-varying simulation results describing the abundance of female mosquitoes per hectare and the disease reproduction number R_0 are made available. Lastly, in the *mosquito_postprocessing* step, the topological proxies (persistence diagrams) on the R_0 maps are generated and deployed.

Figure 14 presents the visualization of the R_0 map with vertical bars highlighting the peaks from the persistence diagram. The simulation result shows a dense region of high R_0 values in the area of Anzio, which was indeed the epicenter of the outbreak in 2017. Also, a high peak in the persistence diagram can be noted in the same area, and this represents a qualitative measure of the overall risk in that subregion. Further peaks in the persistence diagrams are identified by the model in many additional municipalities, including suburbs of Rome, which were subject to significant local disease transmission.

When the workflow is completed, all data is available for the user to download from the VESTEC system for further analysis. This means that results from previously computed incidents can also be overlayed for comparison of different simulation runs.

3) PERFORMANCE

Simulating MBD on a large geographic scale at a high resolution is a computationally demanding problem that can be eased by the use of an HPC machine.

The MBD use-case can exploit two types of parallelism: increasing the number of ensemble simulations and parallelization of a single simulation. To demonstrate the effect of these two ways of parallelizing, we simulate the use case (the chikungunya virus) using a fixed global problem size with 105662 total grid cells, and then we perform two actions: *(i)* we change the number of processor cores allocated to the simulation, and *(ii)* we change the number of ensemble simulations. In both *(i)* and *(ii)*, we record and observe the change in execution time. The results are displayed in Figure 15. We observe that, independently from the number of ensemble simulations, the simulation code can exploit and take advantage of multiple cores to reduce the time to reach a solution. For example, simulating 200 ensembles with a single core takes more than a minute, while using 256 cores in ARCHER2 reduces this time to 2.5 seconds. We also observe that the more ensembles we launch, the better the simulation code scales. For example, the performance of using 10000 ensembles scales much better than only 200 ensembles. These results show that coupling such use-case with HPC resources allows decision-makers to both run simulations faster and get a better understanding of possible countermeasures to cope with such mosquito-borne disease.

4) RELEVANCE TO DECISION MAKERS

The MBD use-case highlights how integrating data-driven epidemiological models with innovative HPC and visualization frameworks can effectively address relevant public health challenges. Model performances were assessed by comparing the obtained epidemiological predictions with known data from previous MBD outbreaks. While we limit the comparison to the chikungunya outbreak in Anzio, the we have also produced valid risk estimates for chikungunya outbreaks in Guardavalle, Italy in 2017, for dengue in Madeira, Portugal in 2012 (mediated by a different vector species), and for Zika in Florida in 2016.

The simulator computes an ensemble of probabilistic estimates on the daily absolute number of adult females per hectare expected for both *Ae. albopictus* and *Ae. aegypti*, estimating the transmission potential for dengue, chikungunya,

and Zika. Model estimates can be obtained for any geographical area of interest, by using only data on local socio-demographic and eco-climatic conditions. Such estimates can be used in several circumstances and are especially useful when there is a lack of entomological and/or epidemiological records in the considered area, or the area to be monitored out-weight the monitoring capability of the public health system. The integration of the developed model with computational technologies provided by other partners in the VESTEC project greatly expands the potential for a useful public health application of the model results. First, the use of HPC systems enabled to speed-up the simulation. Second, the use of interactive and immersive visualization software represents a significant improvement with respect to the current state of the art that relies on static risk maps neglecting the temporal changes associated with the disease risks during the season for public health decision making. Such improvements include three-dimensional map browsing, map search, and the visualization of multiple layers through customizable color schemes and opacity, and filters applied through threshold values defined by the end-user. Finally, the computation of topological proxies provides summary indicators of the risk of infection in different areas. This eases the understanding of the complexity of model outcomes and supports the evaluation of possible countermeasures.

C. CASE STUDY 2: URGENT DECISION MAKING THROUGH UNDERSTANDING OF MAGNETIC RECONNECTION IN SPACE WEATHER

The following section presents the result obtained while executing the Space Weather use-case using the VESTEC system.

1) METHODOLOGY

The overall methodology involves a local terminal node that end-users use to access the space weather workflow, an intermediate visualization node required for in-situ visualization, and the ARCHER2 supercomputer system. There are roughly five steps that are performed in the VESTEC space weather use case: *(i)* a user logs in to the VESTEC system, *(ii)* selects and starts the space weather incident workflow, *(iii)* optionally performs steps to enable *in-situ* visualization of simulations, *(iv)* simulations are carried out using iPIC3D on the supercomputer, and *(v)* after the completion of simulations, the data can be downloaded for offline post-processing (using, e.g., topological proxies).

To better profile and evaluate the performance, this use-case has been executed on a local high-end workstation, available in the consortium, and on the ARCHER2 supercomputer. Since running the workflow is nearly identical between the workstation and the ARCHER2 supercomputer, with the exception of using the visualization node for the catalyst connection, we do not differentiate between them in the qualitative execution section that follows next.

2) EXECUTION

The overall goals of the space weather use-case are: *(i)* to demonstrate the integration and usage of VESTEC components to understand space weather phenomena, *(ii)* to show how supercomputers can help decision-makers make effective decisions regarding threats associated with space weather, and *(iii)* to concretely show an example of how magnetic reconnection can be detected in-situ and with ensemble simulations. The magnetic reconnection simulation can be configured using magnetic field and density values obtained from the NASA MMS spacecrafts' observations [96], and use parameters described in Section IV.

As shown in Figure 16, in *spaceweather_init* step the magnetic reconnection simulation is initialized with a json file containing all the user-defined input parameters. The user has then the option to *in-situ* visualize the (soon to be started) simulation. This is performed by either starting ParaView from the local terminal or from the visualization node on the HPC machine. For example, to *in-situ* visualize in a privately owned cluster machine, VESTEC provides a script that automatically launches several ParaView instances and helps the user to connect them to the simulations. At this point, the user controls one (or several) ParaView instances that are ready to receive data from an active simulation. The user is then enabled to interact with said simulation, for example by pausing and resuming it.

In *spaceweather_simulation* step, the VESTEC system launches and orchestrates several simulations of magnetic reconnection on the supercomputer. The user can either visualize the simulation while it is running or wait until all simulations are completed and then download the final results for an offline inspection. If the user objective is to visualize the simulation, signs of a magnetic reconnection phenomenon can be identified by inspecting the simulation and visualizing the background electron charge density, called $\rho_{e,0}$. For example, Figure 17 shows the contour-plots of $\rho_{e,0}$ for two (of the many) ensemble simulations as they unfold as a function of simulated time. Both these simulations have a similar starting point with minor perturbations in their initial condition, which enable the exploration of possible different outcomes. On the left image of Figure 17, we notice that a magnetic reconnection is occurring since the magnetic field energy is converted into high-speed jets that exit in the positive and negative x-directions.

During the simulation and when the *in-situ* mode is enabled, the user can also choose to analyze any other relevant available metric. For example, in Figure 18, the plane magnetic component (B_z) is shown. In this Figure, we can observe the forming of the quadrupolar structure of the magnetic field. This is caused by the Hall magnetic field and is a clear signature of magnetic reconnection.

Once the user is satisfied with visualizing the simulations and/or the simulations are finished, the data can be downloaded from the VESTEC system for an offline analysis. In *spaceweather_postprocessing* step, the user can analyze the results of many different ensemble simulations that have

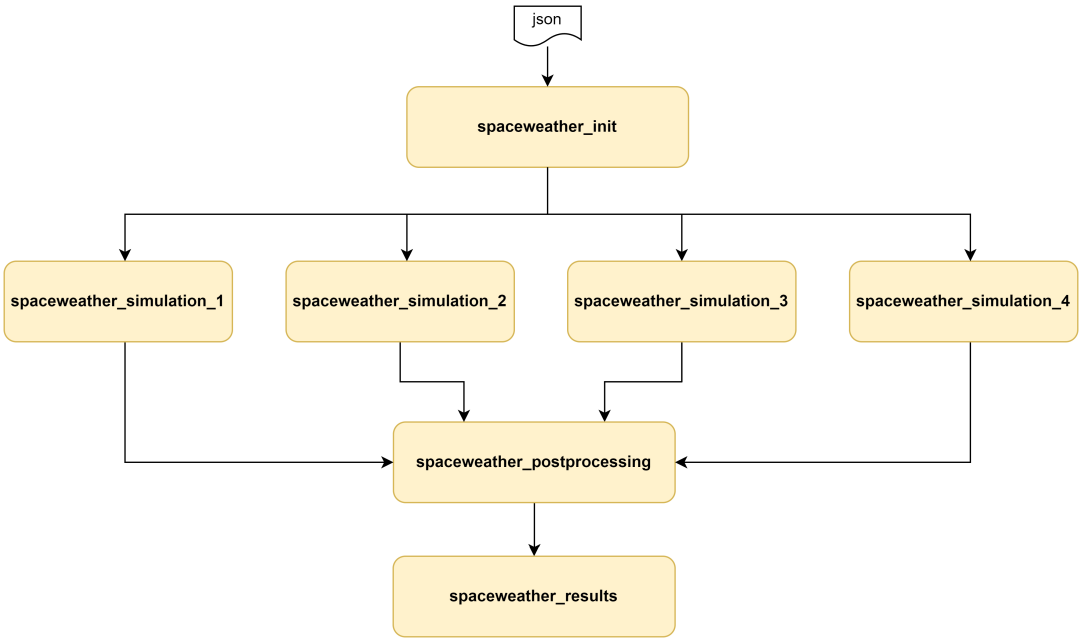


FIGURE 16. Execution workflow for the Space Weather use case.

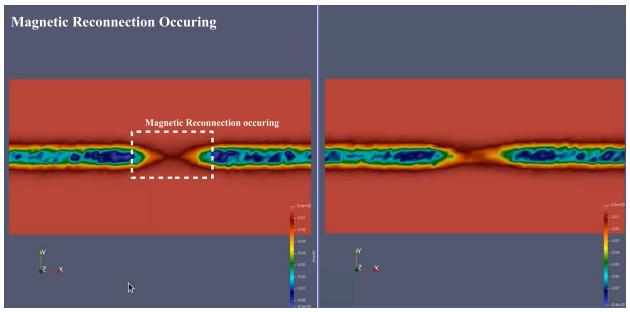


FIGURE 17. Observing the evolution of electron density during magnetic reconnection occurring in-situ using the VESTEC system and the iPIC3D simulator with (right panel) and without (left panel) an initial guide field.

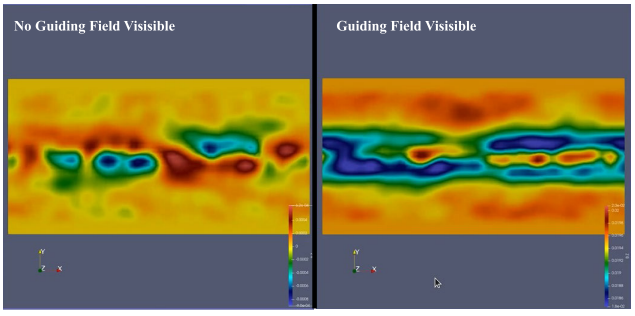


FIGURE 18. Inspecting the contour-plot of out-of-plane magnetic field values (B_z) shows a noticeable guiding field in the right simulation.

been produced during the workflow and summarized with topological proxies. As shown in Figure 19, for example, topological proxies and a dimension reduction technique can be used to generate a point cloud in 2D- or 3D-space.

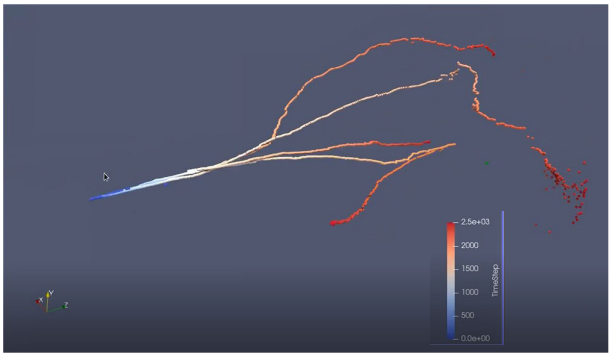


FIGURE 19. Using topological proxies, we can easily visualize the different dynamics of simulations with different guide field values (different trajectories in the topological proxy plot).

Through these techniques, we can track different magnetic reconnection dynamics and detect anomalies, representing regions of interest in the simulation, which can further clarify the simulation results.

Similarly to the MBD use-case, all data generated either in the *simulation* or in the *postprocessing* steps is available for download for further offline analysis.

3) PERFORMANCE

Simulating space weather phenomena is a computationally intensive problem that can benefit the use of an HPC machine. The iPIC3D simulator, used in this use case, has been previously shown to strongly scale to millions of cores, making it ready for future exascale challenges [116], [117]. For this particular use-case, the performance of the system depends on whether Kitware's Catalyst and in-situ visualization is used or not. For example, on ARCHER2, using catalyst leads to complete the entire simulation in ∼546 seconds, while disabling

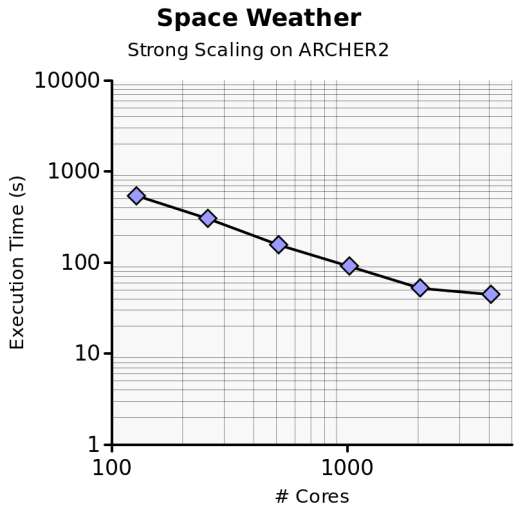


FIGURE 20. The Space Weather use-case strongly scales with the number of available processor cores, here showing that the execution time decrease from ∼9 minutes of simulation time on 128 cores (corresponding to one node) down to ∼45 seconds on 4,096 cores (corresponding to 32 nodes). Such performance enables swift exploration of magnetic reconnection in space and study different scenarios for space weather applications.

Kitware's Catalyst and thus reducing the interactivity lowers the execution time to ∼45 seconds.

Figure 20 shows the scalability of this use case on ARCHER2 system. We monitor how the execution time for 1,000 simulation cycles of the use-case evolves as a function on the number of ARCHER2 nodes used, where one node corresponds to 128 cores. We notice that the speed-up strongly scales when the number of nodes increases. Considering these results, we can claim that in the future, when new HPC systems will be available, the use case problem size can be scaled even with more HPC resources.

4) RELEVANCE TO DECISION MAKERS

The use-case has demonstrated that decision-makers can observe how magnetic reconnection unfolds in-situ under several different realistic scenarios. Magnetic reconnection is part of space weather events that could produce harmful effects for the society and infrastructures in space and on the Earth. Such effects include (but are not limited to): *(i)* atmospheric drag on satellites, which makes them lose altitude (and thus lifespan),[6] *(ii)* overloading and disruption of power grids (caused by induced currents into Earth-bound objects), and, related to the previous one, *(iii)* increased corrosion in oil and natural gas pipelines. Many of the above effects can be mitigated and understood by visually simulating and analyzing space weather phenomena using the presented workflow. One obvious challenge, which is not a limitation in VESTEC per se, is to access space weather data in real-time. In our use-case execution, we used data from the NASA MMS mission, while in the future, we would like to execute the simulations directly on data coming from real-time sensors in orbit (assuming such sensor data is immediately accessible and not released after a long time).

[6]For example, the great solar storm of 1989 caused thousands of space objects to lose several kilometers in altitude [5].

D. CASE STUDY 3: URGENT DECISION MAKING BASED ON FOREST FIRE BEHAVIOUR SIMULATION RESULTS

The following section presents the result obtained while executing the forest fire use-case using the VESTEC system.

1) METHODOLOGY

The forest fire use-case has the scope to trigger and visualize forest fire prediction simulations for a user-defined area of interest (AOI). The goal is to support urgent decision-makers in defining the necessary operational and tactical strategies to fight the fire and determine whether evacuation of certain areas or points in the territory is needed. The use-case allows visualizing and editing hotspot data coming from satellite forest fire sensors (e.g., MODIS and VIIRS sensors). Based on these hotspots, the use-case performs forest fire behavior-related probabilistic simulations through ensemble simulations. From hundreds to thousands of simulations can be triggered and are orchestrated by HPC machine(s) to provide static and probabilistic results to the user through Wildfire Analyst Web.

The overall methodology involves: *(i)* the Wildfire Analyst Web to allow users to access the functionalities of the use-case and to visualize the simulation results, *(ii)* the VESTEC system web interface to monitor the status of the simulation and *(iii)* the ARCHER2 supercomputer system to execute the simulations.

From the user perspective, there are roughly four steps that are performed in the VESTEC system: *(i)* a user logs in to the VESTEC System via the Wildfire Analyst Web interface; then *(ii)* selects simulation parameters (area of interest, ignition, and control points, sets simulation variables configuration), and triggers the wildfire incident workflow in the VESTEC system; *(iii)* simulations are then carried out on the supercomputer, including the computation of topological proxies through TTK; and, lastly, *(iv)* the simulation results are downloaded through a web API interacting integrated in the Wildfire Analyst Web for offline visualization and analysis.

2) EXECUTION

The overall goals of the forest fire use-case were: *(i)* to demonstrate the integration and use of VESTEC components to provide operational and probabilistic forest fire behavior results based on weather, satellite, and user sensor data; *(ii)* to show how supercomputers can help decision-makers to make effective decisions regarding threats associated with forest fires; and *(iii)* to concretely demonstrate examples of fire model estimates using ensemble simulations by leveraging the VESTEC approach.

The forest fire workflow starts when end-users access the Wildfire Analyst Web (see Figure 6). Then, the users can

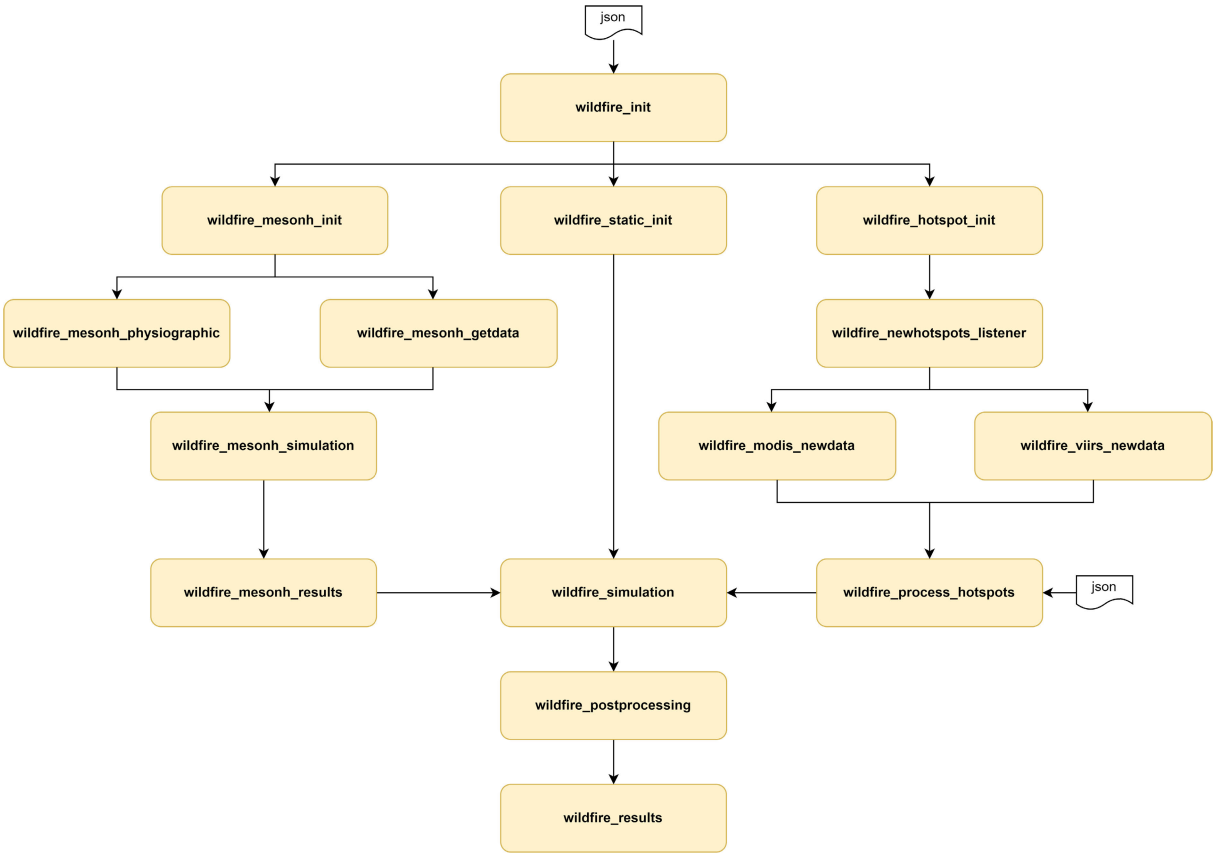


FIGURE 21. Execution workflow for the Wildfire use case.

create an incident in the Wildfire Analyst Web GUI. The incident described in the following is based on one of the worst forest fire ever happened in Spain, the fire in the La Jonquera region in the summer of 2012.

As depicted in Figure 21, the workflow starts with the *wildfire_init* step, in which the user first selects and validates the ignition and control points from the satellite sensor data. Then he/she defines the simulation parameters. The *wildfire_init* step triggers three distinct initialization steps for initializing the MESONH solver, static forest fire data and a hotspot sensors listener (*wildfire_mesonh_init*, *wildfire_static_init* step, and *wildfire_hotspot_init* steps, respectively). When the initilization step is completed, the VESTEC system then orchestrates such ensemble simulations on the HPC machines. The Wildfire Analyst Web and the VESTEC system GUIs keep the user always informed about the progress of simulations.

During the workflow, the VESTEC system uses base maps such as vegetation fuels or DEM layers (*wildfire_mesonh_physiographic* step) and runs the MESO-NH solver to get the latest weather data available for the defined AOI (*wildfire_mesonh_getdata* step). Also, from now on, as the fire progresses, when new sensor data becomes available a new simulation process is launched with updated data. In Figure 21 this corresponds to a loop including the steps between *wildfire_newhotspots_listener*

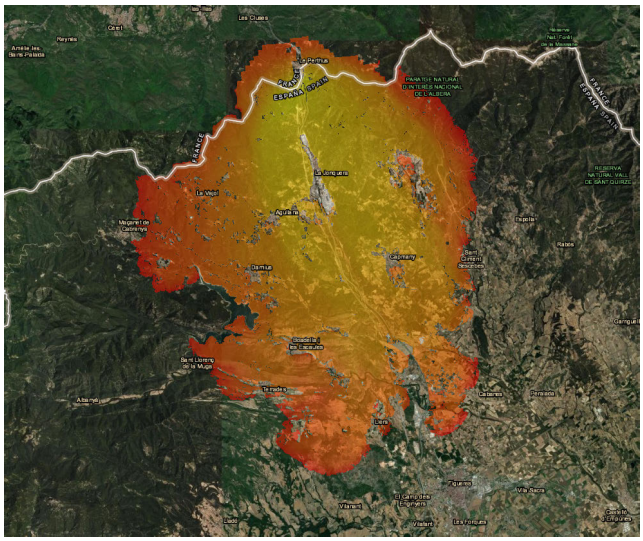


FIGURE 22. Example of a best fit simulation result representing the estimated time of arrival of the fire in Wildfire Analyst.

and *wildfire_process_hotspots*. These sensors can either be hotspots coming from the VIIRS (*wildfire_modis_newdata*) and/or MODIS satellite sensors (*wildfire_viirs_newdata*) (reviewed by the user through ground observations) or new hotspot data entered by the user itself using a JSON text

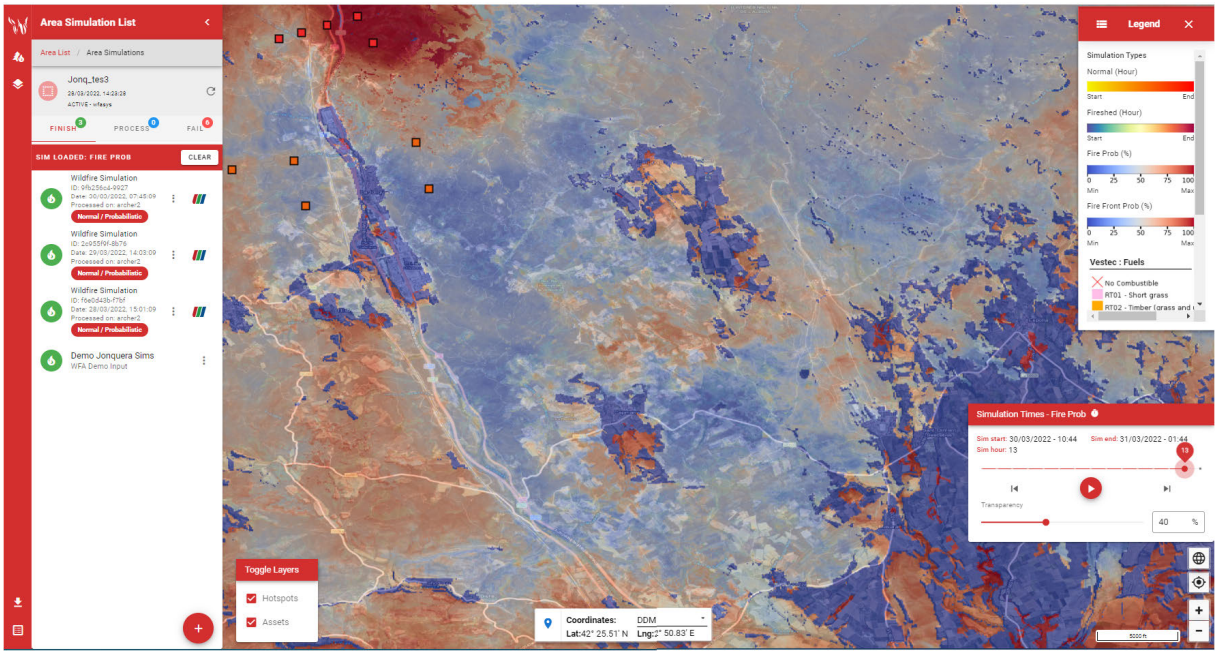


FIGURE 23. An example of the fire presence probability simulation result showing the potential burning probability of a fire in the La Jonquera region (Spain).

file. When MESONH solver and the hotspots complete their tasks, the main wildfire simulation is executed in *wildfire_simulation* step.

The Wildfire Analyst simulation solver generates four main simulation results: *(i)* best static forest fire progression simulation, *(ii)* fire burning probability, *(iii)* fire front presence probability, and *(iv)* fire exposure based on identified assets. The output is encoded in a GeoTIFF raster. The best static forest fire progression simulation is the simulation from the ensemble that best fits the provided control points (selected from VIIRS and MODIS) and the defined simulation parameters and variables (e.g., weather, control points, and fuel conditions weight and deviations). The graphical result of the best static forest fire progression shows the expected arrival time of the fire. The information is displayed to the user through a map in Wildfire Analyst Web.

For example, in Figure 22, yellow represents a lower time of arrival of the fire, while dark orange represents a higher time of arrival. The fire burning probability provides the possibility of fire at each cell. This value is obtained by summing the ensemble simulations. Hence, each cell of the resulting raster includes the burning probability according to the number of times the given cells burned during the Monte Carlo analysis. As shown in Figure 23, this output is represented by using a color scale that represents the ranges of values of the fire presence probability.

The simulation also estimates the probability on the presence of a fire front, and fire exposure sheds around assets, these allow analysing how much time the fire needs to reach a particular user-defined exposure asset. This is based on different conditions such as fuel availability, fuel moisture, terrain topography and the weather. For each probabilistic result each cell in the raster image includes the corresponding probability, i.e., burning probability or fire front location probability.

Additionally to these simulation outputs generated by the solver, in the *wildfire_postprocessing* step, topological descriptors are generated for each simulation. Each simulation ensemble has a set of persistence diagrams that can be further visualized for inspection in ParaView Lite.

3) PERFORMANCE

Simulating fire behavior probabilities is a computationally demanding problem that highly benefits from the use of HPC computing. The WFA model was developed in the VESTEC project, and to estimate the performance of the use-case, we consider simulations at a regional scale. Simulations are performed in parallel and the user can define a number of ensemble members. The result of these ensemble simulations are summarized to fire behavior probabilities, encoded as raster files. In our evaluation, we parallelized over 16 nodes (in total, 1024 cores) on the ARCHER2 system. As expected, the execution time for the computation of individual ensemble simulations decreases over the number of cores used. For simulations that used identical AOI and simulation hours, we observe that simulations having a large deviation in weight parameters (used by a simulation error function defined by the user in the WFA GUI) require more time compared to those using intermediate parameters values (1161 seconds vs. 784.2 seconds).

To compare the ARCHER2 results against those of a local workstation (a workstation with Intel CPU with 4 cores

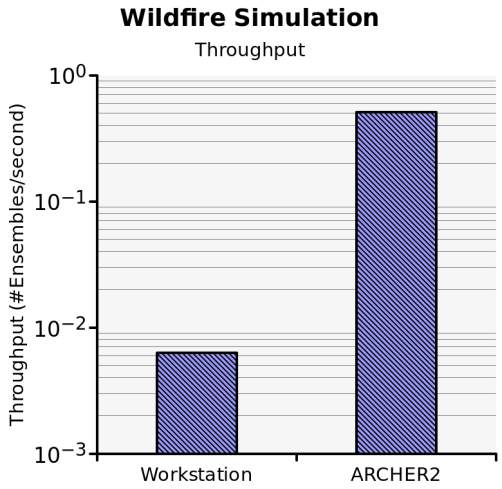


FIGURE 24. The performance impact of coupling High-Performance Computing (HPC) resources with urgent decision making, showing the Wildfire use-case throughput (in ensembles/sec) and comparing a workstation system with the ARCHER2 supercomputer. We see a clear benefit of using supercomputing resources receive better data (more ensembles) to improve decision that can be taken.

and 32 GB of RAM), we benchmark the throughput of ensemble members per second (Figure 24). The HPC system requires 784 seconds (∼13 minutes) to complete the simulation with 400 ensemble members, while the workstation needs 4740 seconds (∼80 minutes) to complete the simulation with only 30 ensemble members.

4) RELEVANCE TO DECISION MAKERS

The use-case execution demonstrated how decision-makers can observe the fire behavior predictions and *(i)* timely act to assign and position firefighting resources to combat the spread of the fire, and *(ii)* take preventive actions to remedy any harmful effects on people and infrastructures inside the affected area. We believe that the ability to (within minutes) be able to easily interpret the visually disseminated forest-fire simulation can drastically support the operational decision and, at the same time, help to save lives and infrastructures and reduce the monetary loss.

Decision-makers can use the results of our simulation to invoke preventive and operational measures, such as identifying high-risk areas (or assets), and, based on the simulation results, to deploy preemptive countermeasures such as vegetation treatments.

VI. CONCLUSION

In this paper, we gave an overview of the VESTEC project [11] and its framework that integrates extreme-scale computing for urgent decision-making. VESTEC has been designed to be used in scenarios where a dangerous or critical event is detected and relayed to a crisis management center that immediately launch, monitor, analyze, and act on a continuously evolving high-precision forecast of the detected critical situation. The main contribution of the VESTEC system has been to bring heterogeneous data into supercomputers, such as data coming from sensor or social networks and/or statistics from the internet, to refine and steer the disaster simulation forecast. VESTEC supports several data analysis methods to extract the most informative pieces of ensemble simulations, which can be visually explored and analyzed. Also, VESTEC supports the injection (performed by the Crisis management staff) of domain-specific knowledge into the system to guide an evolving simulation, so to influence the simulation with the mitigation actions prompted by the crisis manager. Lastly, VESTEC defines a standardized way to integrate new applications or other scenarios through text-based scripting languages (CWL and YAML). For further developments and contributions, the VESTEC System has been released as open source software on GitHub.[7]

In order to demonstrate the scalability and effectiveness of VESTEC system in supporting urgent decision-makers, we have evaluated the system in three different use-cases: *(i)* Mosquito-borne diseases; *(ii)* space weather forecasting; *(iii)* probabilistic forest fire forecast and monitoring. In the experimental evaluation, we demonstrated that HPC can support and improve urgent decision-making processes. VESTEC's flexible and innovative job orchestration and scheduling reduce waiting times and increase utilization. Also, thanks to the efficient parallelization of simulation codes on such machines, the time to get to a solution have been reduced by orders of magnitude. Furthermore, VESTEC enabled the execution of an ensemble of simulations paired with efficient in-situ topological data analysis that permitted to reduce of uncertainties during the decision-making processes. Through the use cases integrated in VESTEC, we have shown the generality of the system. It is then envisioned that users can add their own complex workflows (e.g., for other disasters such as Earthquakes [118]) and tools (e.g., based on deep-learning models such as the recent ChangeOS [119]) extending the VESTEC system.

Therefore, the VESTEC project represents a milestone activity that will drive future effort in the field of extreme-scale computing for urgent decision-making. In the future, further security aspects can be considered, like: *(i)* the certification of crises management centers to request services from dedicated HPC systems, and *(ii)* the protection of the data and communication channel to avoid falsification of streamed results. Also, we can envision the use of Virtual or Mixed Reality for performing real-time mitigation strategies while the simulation is still ongoing.

Systems such as VESTEC are growing in importance, and while there are many activities that are ongoing on the EU level to incorporate HPC resources into urgent decision-making (e.g., ChEESE[8] or LEXIS[9]), there is a dire need to consolidate all these efforts in the future. Since the most important stack-holder is the European Commission, the

[7] https://github.com/VESTEC-EU
[8] https://cheese-coe.eu/
[9] https://lexis-project.eu/web/

responsibility for urgent decision support has to be assigned to governmental agencies. They also have to provide an appropriate infrastructure which can be used immediately and guaranteed as soon as an urgent incident occurs. Consequently, urgent decision making services have to be hosted as well on the EU level. For this reason, Tier-0 HPC centers have to block sufficient computing resources for interactive supercomputing dedicated to emergency cases. But from a crises management center point of view, it should be transparent which computing center is involved. This may lead to cloud-like interfaces.

The current generation of VESTEC system has demonstrated the importance of organising urgent workloads as workflows, and to move forwards we believe that this should be further consolidated. Currently, the workflows in the VESTEC marshalling and control system (powered by RabbitMQ) and those on the HPC machines (powered by CWL) are separate, and it would be beneficial to unify these such that all workflows are described using CWL. This will not only standardise the marshalling and control workflows themselves, but also enable improved error checking. Furthermore, whilst writing these workflows in Python is fairly accessible currently, it would be interesting to explore visual representations of such workflows to lower the barrier to entry further in integration.

Finally, a tight collaboration between experts of different fields is crucial for strengthening the use of HPC technologies in evidence-based decision making.

ACKNOWLEDGMENT

(Markus Flatken, Artur Podobas, and Riccardo Fellegara contributed equally to this work.)

Author Contributions: Andreas Gerndt: leader and coordinator of the VESTEC Project; Markus Flatken and Riccardo Fellegara: design and development of in-memory visual analytics approaches and the extension of the visualization applications; Johannes Holke, David Knapp, Max Kontak, and Achim Basermann: definition and integration of numerical algorithms, data analytics, and optimization approaches in VESTEC system; Michael Nolde and Christian Krullikowski: preparing Earth observation data from space for urgent decision-making as input for simulations; Nick Brown, Rupert Nash, Gordon Gibb, and Evgenij Belikov: background to urgent computing, to the system architecture design, supercomputing orchestration, and integration of high-velocity sensor data into ensemble simulation environments; Artur Podobas, Steven W. D. Chien, and Stefano Markidis: modeling of the space weather use-case; Pierre Guillou, Julien Tierny, and Jules Vidal: defining topological data analysis strategies for extracting features and interactively explore the data; Charles Gueunet: supporting data post-processing, feature extraction, and visualization of sensor data; Johannes Günther and Miroslaw Pawlowski: design and adaptions of the visualization pipelines; Piero Poletti, Giorgio Guzzetta, Mattia Manica, and Agnese Zardini: modeling of the mosquito borne diseases use-case; Jean-Pierre Chaboureau: preparation and understanding of weather data; and Miguel Mendes, Adrián Cardil, Santiago Monedero, and Joaquin Ramirez: modeling of the Wildfire use-case.

REFERENCES

[1] A. Taylor. (Jul. 2017). *Wildfires Across Southern Europe*. [Online]. Available: https://www.theatlantic.com/photo/2017/07/wildfires-across-southern-europe/534039/

[2] (2021). *National Hurrican Center*. [Online]. Available: https://www.nhc.noaa.gov/data/tcr/

[3] J. Watts. (Jul. 2021). *Climate Scientists Shocked by Scale of Floods in Germany*. [Online]. Available: https://www.theguardian.com/environment/2021/jul/16/climate-scientists-shocked-by-scale-of-floods-in-germany

[4] L. A. Reperant and A. D. M. E. Osterhaus, "AIDS, avian flu, SARS, MERS, ebola, Zika··· what next?" *Vaccine*, vol. 35, no. 35, pp. 4470–4474, Aug. 2017.

[5] V. Jordanova, G. Delzanno, M. Henderson, H. Godinez, C. Jeffery, E. Lawrence, S. Morley, J. Moulton, L. Vernon, J. Woodroffe, T. Brito, M. Engel, C. Meierbachtol, D. Svyatsky, Y. Yu, G. Tóth, D. Welling, Y. Chen, J. Haiducek, S. Markidis, J. Albert, J. Birn, M. Denton, and R. Horne, "Specification of the near-Earth space environment with SHIELDS," *J. Atmos. Solar-Terr. Phys.*, vol. 177, pp. 148–159, Oct. 2018.

[6] J. Ramirez, S. Monedero, and D. Buckley, "New approaches in fire simulations analysis with wildfire analyst," in *Proc. 5th Int. Wildland Fire Conf.*, 2011, pp. 9–13.

[7] R. Rothermel, *A Mathematical Model for Predicting Fire Spread in Wildland Fuels*, vol. 115. Washington, DC, USA: USDA Forest Service, 1972.

[8] M. Yu, C. Yang, and Y. Li, "Big data in natural disaster management: A review," *Geosciences*, vol. 8, no. 5, p. 165, May 2018.

[9] O. M. Araz, T. Choi, D. L. Olson, and F. S. Salman, "Role of analytics for operational risk management in the era of big data," *Decis. Sci.*, vol. 51, no. 6, pp. 1320–1346, Dec. 2020.

[10] S. Jeble, S. Kumari, V. G. Venkatesh, and M. Singh, "Influence of big data and predictive analytics and social capital on performance of humanitarian supply chain: Developing framework and future research directions," *Benchmarking: Int. J.*, vol. 27, no. 2, pp. 606–633, Mar. 2020.

[11] European Commission Grant H2020-FETHPC-2017. (2022). *Visual Exploration and Sampling Toolkit for Extreme Computing (VESTEC)*. [Online]. Available: https://cordis.europa.eu/project/id/800904

[12] P. Beckman, S. Nadella, N. Trebon, and I. Beschastnikh, "SPRUCE: A system for supporting urgent high-performance computing," in *Grid-Based Problem Solving Environments*. Cham, Switzerland: Springer, 2007, pp. 295–311.

[13] J. Ahrens, B. Geveci, and C. Law, "ParaView: An end-user tool for large-data visualization," in *The Visualization Handbook*. Burlington, MA, USA: Elsevier, 2005, pp. 717–731.

[14] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, and P. Navrátil, "VisIt: An end-user tool for visualizing and analyzing very large data," in *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*. Boca Raton, FL, USA: CRC Press, Oct. 2012, pp. 357–372.

[15] W. J. Schroeder, K. Martin, W. E. Lorensen, L. S. Avila, and K. W. Martin, *The Visualization Toolkit*, 4th ed. Clifton Park, NY, USA: Kitware, 2006.

[16] E. W. Bethel, H. Childs, and C. D. Hansen, *High Performance Visualization: Enabling Extreme-Scale Scientific Insight* (Chapman & Hall/CRC Computational Science Series). Boca Raton, FL, USA: CRC Press, 2012.

[17] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, and K. E. Jansen, "The ParaView coprocessing library: A scalable, general purpose in situ visualization library," in *Proc. IEEE Symp. Large Data Anal. Visualizat.*, Oct. 2011, pp. 89–96.

[18] U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian, and J. Mauldin, "ParaView Catalyst: Enabling in situ data analysis and visualization," in *Proc. 1st Workshop In Situ Infrastructures Enabling Extreme-Scale Anal. Visualizat.* New York, NY, USA: Association for Computing Machinery, 2015, pp. 25–29.

[19] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux, "The topology toolkit," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 832–842, Jan. 2018.

[20] T. Bin Masood, J. Budin, M. Falk, G. Favelier, C. Garth, C. Gueunet, P. Guillou, L. Hofmann, P. Hristov, A. Kamakshidasan, C. Kappe, P. Klacansky, P. Laurin, J. Levine, J. Lukasczyk, D. Sakurai, M. Soler, P. Steneteg, J. Tierny, W. Usher, J. Vidal, and M. Wozniak, "An overview of the topology toolkit," in *TopoInVis*. Cham, Switzerland: Springer, 2019.

[21] Wald, I, G. Johnson, J. Amstutz, C. Brownlee, A. Knoll, J. Jeffers, J. Günther, and P. Navratil, "OSPRay—A CPU ray tracing framework for scientific visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 931–940, Jan. 2017.

[22] W. Usher, I. Wald, J. Amstutz, J. Günther, C. Brownlee, and V. Pascucci, "Scalable ray tracing using the distributed FrameBuffer," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 455–466, Jun. 2019.

[23] H. Edelsbrunner and J. Harer, *Computational Topology: An Introduction*. Providence, RI, USA: American Mathematical Society, 2009.

[24] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth, "A survey of topology-based methods in visualization," *Comput. Graph. Forum*, vol. 35, no. 3, pp. 643–667, Jun. 2016.

[25] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci, "Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 5, pp. 1053–1060, Sep. 2006.

[26] P.-T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. Bell, "Interactive exploration and analysis of large-scale simulations using topology-based data segmentation," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 9, pp. 1307–1324, Sep. 2011.

[27] A. Gyulassy, P. T. Bremer, R. Grout, H. Kolla, J. Chen, and V. Pascucci, "Stability of dissipation elements: A case study in combustion," *Comput. Graph. Forum*, vol. 33, no. 3, pp. 51–60, Jun. 2014.

[28] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann, "Topologically clean distance fields," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1432–1439, Nov. 2007.

[29] A. Gyulassy, A. Knoll, K. C. Lau, B. Wang, P.-T. Bremer, M. E. Papka, L. A. Curtiss, and V. Pascucci, "Interstitial and interlayer ion diffusion geometry extraction in graphitic nanosphere battery materials," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 916–925, Jan. 2016.

[30] G. Favelier, C. Gueunet, and J. Tierny, "Visualizing ensembles of viscous fingers," in *Proc. IEEE SciVis Contest*, 2016, pp. 1–18.

[31] M. Soler, M. Petitfrere, G. Darche, M. Plainchault, B. Conche, and J. Tierny, "Ranking viscous finger simulations to an acquired ground truth with topology-aware matchings," in *Proc. IEEE 9th Symp. Large Data Anal. Visualizat. (LDAV)*, 2019, pp. 62–72.

[32] D. Maljovec, B. Wang, P. Rosen, A. Alfonsi, G. Pastore, C. Rabiti, and V. Pascucci, "Topology-inspired partition-based sensitivity analysis and visualization of nuclear simulations," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, 2016, pp. 64–71.

[33] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege, "Two-dimensional time-dependent vortex regions based on the acceleration magnitude," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2080–2087, Dec. 2011.

[34] K. Anderson, J. Anderson, S. Palande, and B. Wang, "Topological data analysis of functional MRI connectivity in time and space domains," in *Proc. Int. Workshop Connectomics Neuroimaging*. Cham, Switzerland: Springer, 2018, pp. 67–77.

[35] D. Günther, R. A. Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny, "Characterizing molecular interactions in chemical systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2476–2485, Dec. 2014.

[36] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P. Bremer, "TopoMS : Comprehensive topological exploration for molecular and condensed-matter systems," *J. Comput. Chem.*, vol. 39, no. 16, pp. 936–952, Jun. 2018.

[37] M. Olejniczak, A. Severo Pereira Gomes, and J. Tierny, "A topological data analysis perspective on noncovalent interactions in relativistic calculations," *Int. J. Quantum Chem.*, vol. 120, no. 8, Apr. 2020, Art. no. e26133.

[38] T. Sousbie, "The persistent cosmic web and its filamentary structure: Theory and implementations," *Roy. Astronomical Soc.*, vol. 414, no. 1, pp. 350–383, 2011.

[39] N. Shivashankar, P. Pranav, V. Natarajan, R. v. d. Weygaert, E. G. P. Bos, and S. Rieder, "Felix: A topology based framework for visual exploration of cosmic filaments," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 6, pp. 1745–1759, Jun. 2016.

[40] Edelsbrunner, Letscher, and Zomorodian, "Topological persistence and simplification," *Discrete Comput. Geometry*, vol. 28, no. 4, pp. 511–533, Nov. 2002.

[41] H. Edelsbrunner, D. Morozov, and V. Pascucci, "Persistence-sensitive simplification functions on 2-manifolds," in *Proc. 22nd Annu. Symp. Comput. Geometry*, Jun. 2006, pp. 127–134.

[42] J. Tierny and V. Pascucci, "Generalized topological simplification of scalar fields on surfaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2005–2013, Dec. 2012.

[43] J. Lukasczyk, C. Garth, R. Maciejewski, and J. Tierny, "Localized topological simplification of scalar data," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 572–582, Feb. 2021.

[44] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," in *Proc. 11th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2000, pp. 918–926.

[45] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny, "Contour forests: Fast multi-threaded augmented contour trees," in *Proc. IEEE 6th Symp. Large Data Anal. Visualizat. (LDAV)*, Oct. 2016, pp. 85–92.

[46] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny, "Task-based augmented merge trees with Fibonacci heaps," in *Proc. IEEE 7th Symp. Large Data Anal. Visualizat. (LDAV)*, Oct. 2017, pp. 6–15.

[47] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny, "Task-based augmented contour trees with Fibonacci heaps," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1889–1905, Aug. 2019.

[48] V. Pascucci, G. Scorzelli, P. T. Bremer, and A. Mascarenhas, "Robust on-line computation of Reeb graphs: Simplicity and speed," *ACM Trans. Graph.*, vol. 26, no. 3, p. 58, 2007.

[49] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno, "Reeb graphs for shape analysis and applications," *Theor. Comput. Sci.*, vol. 392, nos. 1–3, pp. 5–22, Feb. 2008.

[50] J. Tierny, A. Gyulassy, and E. Simon, "Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 1177–1184, Nov. 2009.

[51] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny, "Task-based augmented Reeb graphs with dynamic ST-trees," in *Proc. Eurographics Symp. Parallel Graph. Visualizat.*, 2019, pp. 1–11.

[52] A. Gyulassy, P.-T. Bremer, B. Hamann, and V. Pascucci, "A practical approach to morse-smale complex computation: Scalability and generality," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1619–1626, Nov. 2008.

[53] A. Gyulassy, D. Günther, J. A. Levine, J. Tierny, and V. Pascucci, "Conforming morse-smale complexes," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2595–2603, Dec. 2014.

[54] L. De Floriani, U. Fugacci, F. Iuricich, and P. Magillo, "Morse complexes for shape segmentation and homological analysis: Discrete models and algorithms," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 761–785, 2015.

[55] A. Gyulassy, P.-T. Bremer, and V. Pascucci, "Shared-memory parallel computation of morse-smale complexes with improved accuracy," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 1183–1192, Jan. 2019.

[56] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," in *Proc. 21st Annu. Symp. Comput. Geometry*, Jun. 2005, pp. 263–271.

[57] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4741–4748.

[58] M. Carriére, M. Cuturi, and S. Oudot, "Sliced Wasserstein kernel for persistence diagrams," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 664–673.

[59] B. Rieck, F. Sadlo, and H. Leitte, "Topological machine learning with persistence indicator functions," in *Topological Methods in Data Analysis and Visualization*. Cham, Switzerland: Springer, 2017, pp. 87–101.

[60] R. Cotsakis, J. Shaw, J. Tierny, and J. A. Levine, "Implementing persistence-based clustering of point clouds in the topology toolkit," in *Topological Methods in Data Analysis and Visualization VI*. Cham, Switzerland: Springer, 2021, pp. 343–357.

[61] H. Doraiswamy, J. Tierny, P. J. S. Silva, L. G. Nonato, and C. Silva, "TopoMap: A 0-dimensional homology preserving projection of high-dimensional data," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 561–571, Feb. 2021.

[62] J. Vidal, J. Budin, and J. Tierny, "Progressive Wasserstein barycenters of persistence diagrams," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 151–161, Jan. 2020.
[63] L. Kantorovich, "On the translocation of masses," in *Proc. AS URSS*, 1942, pp. 227–229.
[64] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Académie Royale des Sciences de Paris*, 1781.
[65] K. Turner, Y. Mileyko, S. Mukherjee, and J. Harer, "Fréchet means for distributions of persistence diagrams," *Discrete Comput. Geometry*, vol. 52, no. 1, pp. 44–70, Jul. 2014.
[66] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, Mar. 1957.
[67] D. P. Bertsekas, "A new algorithm for the assignment problem," *Math. Program.*, vol. 21, no. 1, pp. 152–171, Dec. 1981.
[68] M. Kerber, D. Morozov, and A. Nigmetov, "Geometry helps to compare persistence diagrams," *ACM J. Experim. Algorithmics*, vol. 22, pp. 1–20, Dec. 2017.
[69] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Comput.*, vol. 10, no. 6, pp. 87–89, Nov. 2006.
[70] L. Johansson and D. Dossot, *RabbitMQ Essentials: Build Distributed and Scalable Applications With Message Queuing Using RabbitMQ*. Birmingham, U.K.: Packt Publishing, 2020.
[71] G. P. Gibb, N. Brown, R. W. Nash, M. Mendes, S. Monedero, H. D. Fidalgo, J. R. Cisneros, A. Cardil, and M. Kontak, "A bespoke workflow management system for data-driven urgent HPC," in *Proc. IEEE/ACM HPC Urgent Decis. Making (UrgentHPC)*, 2020, pp. 10–20.
[72] G. Gibb, R. Nash, N. Brown, and B. Prodan, "The technologies required for fusing HPC and real-time data to support urgent computing," in *Proc. IEEE/ACM HPC Urgent Decis. Making (UrgentHPC)*, Nov. 2019, pp. 24–34.
[73] M. R. Crusoe, S. Abeln, A. Iosup, P. Amstutz, J. Chilton, N. Tijanić, H. Ménager, S. Soiland-Reyes, B. Gavrilovic, and C. Goble, "Methods included: Standardizing computational reuse and portability with the common workflow language," 2021, *arXiv:2105.07028*.
[74] R. W. Nash, M. R. Crusoe, M. Kontak, and N. Brown, "Supercomputing with MPI meets the common workflow language standards: An experience report," in *Proc. IEEE/ACM Workflows Support Large-Scale Sci. (WORKS)*, Nov. 2020, pp. 17–24.
[75] M. Soler, M. Plainchault, B. Conche, and J. Tierny, "Topologically controlled lossy compression," in *Proc. IEEE Pacific Visualizat. Symp. (PacificVis)*, Apr. 2018, pp. 46–55.
[76] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers, and M. Petersen, "An image-based approach to extreme scale in situ visualization and analysis," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2014, pp. 424–434.
[77] J. Kruskal and M. Wish, *Multidimensional Scaling*. Thousand Oaks, CA, USA: Sage University Publications, 1978.
[78] J. Vidal, P. Guillou, and J. Tierny, "A progressive approach to scalar field topology," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 6, pp. 2833–2850, Jun. 2021.
[79] M. Kontak, J. Vidal, and J. Tierny, "Statistical parameter selection for clustering persistence diagrams," in *Proc. IEEE/ACM HPC Urgent Decis. Making (UrgentHPC)*, 2019, pp. 7–12.
[80] S. Schneegans, M. Flatken, and A. Gerndt. (Mar. 2021). *Cosmoscout VR 1.4.0*. [Online]. Available: https://doi.org/10.5281/zenodo.4646924
[81] S. Schneegans, M. Zeumer, J. Gilg, and A. Gerndt, "CosmoScout VR: A modular 3D solar system based on SPICE," in *Proc. IEEE Aerosp. Conf. (AERO)*, Mar. 2022, pp. 1–13.
[82] G. G. Brown and A. L. Vassiliou, "Optimizing disaster relief: Real-time operational and tactical decision support," *Nav. Res. Logistics*, vol. 40, no. 1, pp. 1–23, Feb. 1993.
[83] S. Wu, L. Shuman, B. Bidanda, O. Prokopyev, M. Kelley, K. Sochats, and C. Balaban, "Simulation-based decision support system for real-time disaster response management," in *Proc. IIE Annual Conf.* Peachtree Corners, GA, USA: Institute of Industrial and Systems Engineers (IISE), 2008, p. 58.
[84] M. Khouj, C. López, S. Sarkaria, and J. Marti, "Disaster management in real time simulation using machine learning," in *Proc. 24th Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2011, pp. 001507–001510.
[85] S. Koshimura, "Establishing the advanced disaster reduction management system by fusion of real-time disaster simulation and big data assimilation," *J. Disaster Res.*, vol. 11, no. 2, pp. 164–174, Mar. 2016.
[86] A. Musa, O. Watanabe, H. Matsuoka, H. Hokari, T. Inoue, Y. Murashima, Y. Ohta, R. Hino, S. Koshimura, and H. Kobayashi, "Real-time tsunami inundation forecast system for tsunami disaster prevention and mitigation," *J. Supercomput.*, vol. 74, no. 7, pp. 3093–3113, Jul. 2018.
[87] Centers for Disease Control. (2019). *Epidemic Prediction Initiative—AEDES Forecasting*. [Online]. Available: https://predict.cdc.gov/post/5c4f6d687620e103b6dcd015
[88] Q. Zhang, K. Sun, M. Chinazzi, A. P. Y. Piontti, N. Dean, and D. Rojas, "Spread of Zika virus in the Americas," *Proc. Nat. Acad. Sci. USA*, vol. 114, pp. E4334–E4343, May 2017.
[89] M. Yamada, R. Kulsrud, and H. Ji, "Magnetic reconnection," *Rev. Modern Phys.*, vol. 82, no. 1, p. 603, 2010.
[90] G. Lapenta, V. Pierrard, R. Keppens, S. Markidis, S. Poedts, O. Šebek, P. M. Trávníček, P. Henri, F. Califano, F. Pegoraro, M. Faganello, V. Olshevsky, A. L. Restante, Å. Nordlund, J. T. Frederiksen, D. H. Mackay, C. E. Parnell, A. Bemporad, R. Susino, and K. Borremans, "SWIFF: Space weather integrated forecasting framework," *J. Space Weather Space Climate*, vol. 3, p. A05, May 2013.
[91] S. Markidis, V. Olshevsky, G. Tóth, Y. Chen, I. B. Peng, G. Lapenta, and T. Gombosi, "Kinetic modeling in the magnetosphere," in *Magnetospheres in the Solar System*. Hoboken, NJ, USA: Wiley, 2021, pp. 607–615.
[92] I. B. Peng, S. Markidis, A. Vaivads, J. Vencels, J. Amaya, A. Divin, E. Laure, and G. Lapenta, "The formation of a magnetosphere with implicit particle-in-cell simulations," *Proc. Comput. Sci.*, vol. 51, pp. 1178–1187, 2015.
[93] Y. V. Khotyaintsev, A. Divin, A. Vaivads, M. André, and S. Markidis, "Energy conversion at dipolarization fronts," *Geophys. Res. Lett.*, vol. 44, no. 3, pp. 1234–1242, Feb. 2017.
[94] S. Markidis, G. Lapenta, and Rizwan-Uddin, "Multi-scale simulations of plasma with iPIC3D," *Math. Comput. Simul.*, vol. 80, no. 7, pp. 1509–1519, Mar. 2010.
[95] J. Birn, J. F. Drake, M. A. Shay, B. N. Rogers, R. E. Denton, M. Hesse, M. Kuznetsova, Z. W. Ma, A. Bhattacharjee, A. Otto, and P. L. Pritchett, "Geospace environmental modeling (GEM) magnetic reconnection challenge," *J. Geophys. Res., Space Phys.*, vol. 106, no. A3, pp. 3715–3719, Mar. 2001.
[96] J. L. Burch, T. E. Moore, R. B. Torbert, and B. L. Giles, "Magnetospheric multiscale overview and science objectives," *Space Sci. Rev.*, vol. 199, nos. 1–4, pp. 5–21, Mar. 2016.
[97] S. Markidis, I. Peng, A. Podobas, I. Jongsuebchoke, G. Bengtsson, and P. Herman, "Automatic particle trajectory classification in plasma simulations," in *Proc. IEEE/ACM Workshop Mach. Learn. High Perform. Comput. Environments (MLHPC) Workshop Artif. Intell. Mach. Learn. Sci. Appl. (AIS)*, Nov. 2020, pp. 64–71.
[98] S. Markidis, P. Henri, G. Lapenta, K. Rönnmark, M. Hamrin, Z. Meliani, and E. Laure, "The fluid-kinetic particle-in-cell method for plasma simulations," *J. Comput. Phys.*, vol. 271, pp. 415–429, Aug. 2014.
[99] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*. Boca Raton, FL, USA: CRC Press, 2018.
[100] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*. Boca Raton, FL, USA: CRC Press, 1988.
[101] S. Markidis and G. Lapenta, "The energy conserving particle-in-cell method," *J. Comput. Phys.*, vol. 230, no. 18, pp. 7037–7052, Aug. 2011.
[102] S. Markidis, V. Olshevsky, C. P. Sishtla, S. W. D. Chien, E. Laure, and G. Lapenta, "PolyPIC: The polymorphic-particle-in-cell method for fluid-kinetic coupling," *Frontiers Phys.*, vol. 6, p. 100, Oct. 2018.
[103] C. P. Sishtla, S. W. D. Chien, V. Olshevsky, E. Laure, and S. Markidis, "Multi-GPU acceleration of the iPIC3D implicit particle-in-cell code," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2019, pp. 612–618.
[104] S. W. Chien, J. Nylund, G. Bengtsson, I. B. Peng, A. Podobas, and S. Markidis, "sputniPIC: An implicit particle-in-cell code for multi-GPU systems," in *Proc. IEEE 32nd Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, 2020, pp. 149–156.
[105] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986.
[106] F. Albini, *Estimating Wildfire Behavior and Effects*, vol. 30. Washington, DC, USA: USDA Forest Service, 1976.
[107] C. Lac et al., "Overview of the MESO-NH model version 5.4 and its applications," *Geoscientific Model Develop.*, vol. 11, no. 5, pp. 1929–1969, 2018.

[108] F. Pantillon, P. Mascart, J.-P. Chaboureau, C. Lac, J. Escobar, and J. Duron, "Seamless MESO-NH modeling over very large grids," *Comp. Rendus Mécanique*, vol. 339, nos. 2–3, pp. 136–140, Feb. 2011.
[109] T. Dauhut, J. Chaboureau, J. Escobar, and P. Mascart, "Large-eddy simulations of Hector the convector making the stratosphere wetter," *Atmos. Sci. Lett.*, vol. 16, no. 2, pp. 135–140, Apr. 2015.
[110] M. A. Finney, "Fire growth using minimum travel time methods," *Can. J. Forest Res.*, vol. 32, no. 8, pp. 1420–1424, Aug. 2002.
[111] A. Cardil, S. Monedero, C. A. Silva, and J. Ramirez, "Adjusting the rate of spread of fire simulations in real-time," *Ecological Model.*, vol. 395, pp. 39–44, Mar. 2019.
[112] R. Rothermel and G. Rinehart, *Field Procedures for Verification and Adjustment of Fire Behavior Predictions*, no. 142. Washington, DC, USA: USDA Forest Service, 1983.
[113] T. J. Cova, P. E. Dennison, T. H. Kim, and M. A. Moritz, "Setting wildfire evacuation trigger points using fire spread modeling and GIS," *Trans. GIS*, vol. 9, no. 4, pp. 603–617, Oct. 2005.
[114] S. Monedero, J. Ramírez, D. Molina-Terrén, and A. Cardil, "Simulating wildfires backwards in time from the final fire perimeter in point-functional fire models," *Environ. Model. Softw.*, vol. 92, pp. 163–168, Jun. 2017.
[115] S. E. Fick and R. J. Hijmans, "WorldClim 2: New 1-km spatial resolution climate surfaces for global land areas," *Int. J. Climatol.*, vol. 37, no. 12, pp. 4302–4315, Oct. 2017.
[116] M. E. Innocenti, A. Johnson, S. Markidis, J. Amaya, J. Deca, V. Olshevsky, and G. Lapenta, "Progress towards physics-based space weather forecasting with exascale computing," *Adv. Eng. Softw.*, vol. 111, pp. 3–17, Sep. 2017.
[117] G. Lapenta, S. Markidis, S. Poedts, and D. Vucinic, "Space weather prediction and exascale computing," *Comput. Sci. Eng.*, vol. 15, no. 5, pp. 68–76, Sep. 2013.
[118] A. Tang and A. Wen, "An intelligent simulation system for earthquake disaster assessment," *Comput. Geosci.*, vol. 35, no. 5, pp. 871–879, May 2009.
[119] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, "Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters," *Remote Sens. Environ.*, vol. 265, Nov. 2021, Art. no. 112636.

MARKUS FLATKEN received the M.Sc. degree from the Hannover University of Applied Sciences and Arts, in 2009. He has been a Research Scientist with the German Aerospace Center, Institute for Software Technology, since 2010. His research interests include scientific visualization, computer graphics, and high-performance visualization with the Department of Software for Space Systems and Interactive Visualization. He works on the development of software frameworks and applications for large-scale data visualization and analysis, utilizing high-performance computing techniques, and resources.

ARTUR PODOBAS received the Ph.D. degree from the KTH Royal Institute of Technology, Sweden, in 2015. He is currently an Assistant Professor with the KTH Royal Institute of Technology. Between 2015 and 2016, he was a Postdoctoral Fellow with Denmark Technical University, DTU Compute. Between 2016 and 2018, he was a Japan Society of Promotional of Science (JSPS) Fellow of GSIC, Tokyo Institute of Technology, Japan. From 2019 to 2020, he was a Postdoctoral Researcher with the Processor Research Team, RIKEN, Japan. His research interests include high-performance accelerator research, which includes reconfigurable architectures (FPGAs, CGRAs), neuromorphic computing, parallel architectures, and high-level synthesis for high-performance systems (HPC).

RICCARDO FELLEGARA received the master's and Ph.D. degrees in computer science from the University of Genoa, Italy, in 2010 and 2015, respectively. He is currently a Research Scientist with the Department of Software for Space Systems and Interactive Visualization, German Aerospace Center (DLR), Institute for Software Technology, Germany. He has been a Postdoctoral Fellow Affiliate with the Computer Science Department and Geographical Sciences Department, University of Maryland, College Park, USA, from 2015 to 2019. His research interests include spatial data structures and algorithms, scientific visualization, topology-based data analysis, high performance computing (HPC), geometric modeling, and geographic information systems.

ACHIM BASERMANN received the Dr.-Ing. degree, and the Ph.D. degree in electrical engineering from RWTH Aachen, in 1995. He has a postdoctoral position in computer science with Research Centre Jülich GmbH, Central Institute for Applied Mathematics. He is currently the Head of the Department of High-Performance Computing, German Aerospace Center (DLR), Simulation and Software Technology Institute, and a German Research Foundation (DFG) Review Board Member of computer science, topic "Massively Parallel and Data Intensive Systems." He coordinated the application workpackage in the European Grid Computing Project NextGRID (2004–2007), the pre- and postprocessing activities in the European Exascale Computing Project CRESTA (2011–2014), and the algorithmic research in the Exascale Computing Projects ESSEX I and ESSEX II of DFG (2013–2018). From 1997 to 2009, he led a team of HPC application experts with the C&C Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany. His current research interests include massively parallel linear algebra algorithms, partitioning methods, optimization tools in the area of computational fluid dynamics for many-core architectures and GPGPU clusters, high-performance data analytics, and quantum computing.

JOHANNES HOLKE received the Ph.D. degree. He is currently a Research Group Leader of Scalable Adaptive Mesh Refinement with the High-performance Computing Department, DLR Institute for Software Technology. He is the lead author of the massively scalable adaptive mesh refinement library t8code. His research interests include developing and optimizing parallel algorithms for modern supercomputer architectures.

DAVID KNAPP is currently pursuing the Ph.D. degree with the University of Cologne. He is also a Researcher with the High Performance Computing Department, Germane Aerospace Center. Besides HPC, he is interested in adaptive mesh refinement and in discontinuous Galerkin methods.

MAX KONTAK received the Ph.D. degree in mathematics from the University of Siegen, in 2018. His Ph.D. thesis is about the efficient numerical solution of inverse problems. From 2018 to 2020, he was a Researcher with the Institute for Software Technology, DLR German Aerospace Center, working on helicopter simulation and the Horizon 2020 Project VESTEC. Since 2021, he has been pursuing other opportunities outside academia as a Data Scientist.

CHRISTIAN KRULLIKOWSKI received the B.S. degree in geological sciences and the M.S. degree in geological sciences with a specialization in hydrogeology from Freie Universität Berlin, Germany, in 2013 and 2016, respectively.

In 2017, he joined the German Remote Sensing Data Center (DFD), German Aerospace Center (DLR), Department Geo-Risks and Civil Security. His research interests include thematic processing with rule-based and machine learning methods for crisis-related information extraction from optical, thermal, and SAR remote sensing imagery.

MICHAEL NOLDE received the Ph.D. degree from Kiel University, Germany, in 2013. His Ph.D. dissertation was titled, ''Development of an Automated Forecasting Platform for Wildfire Danger on the Isle of Sardinia/Italy.'' He is currently a Researcher with the German Aerospace Center (DLR), German Remote Sensing Data Center (DFD), Department of Geo-Risks and Civil Security. He is focused on the analysis of wildfire events using satellite data. Later on, he held a postdoctoral position with the Centro Euro-Mediteraneo sui Cambiamenti Climatici, Lecce, Italy. At his current position, he is responsible for developing and maintaining automatic processing chains for extracting information from satellite data in the context of crisis information.

NICK BROWN received the Ph.D. degree. He is currently a Research Fellow with EPCC, The University of Edinburgh, with interests in parallel programming abstractions, compilers, novel architectures, and HPC. He led the interactive HPC work-package on the VESTEC EU FET project, which was concerned with interactive supercomputing for urgent workloads and he has chaired the UrgentHPC workshops at recent SC conferences.

RUPERT NASH received the Ph.D. degree. He is currently a Research Fellow with EPCC, The University of Edinburgh. His research interest includes high performance computing for fluid dynamics applications, particularly using lattice Boltzmann methods. He is also interested in the use of workflows and other tools to enable wider use of HPC applications and systems.

GORDON GIBB received the Ph.D. degree. He is currently a Senior Research Developer with the Institute for Astronomy, School of Physics, The University of Edinburgh. With experience in HPC code development and parallelization, he was involved in developing the marshaling and control workflows of the VESTEC urgent computing project when he worked for EPCC.

EVGENIJ BELIKOV is currently an Applications Consultant in high-performance computing with EPCC, The University of Edinburgh. His research interests include parallel programming models, exascale computing, scheduling, and performance optimization.

STEVEN W. D. CHIEN received the Ph.D. degree in computer science from the KTH Royal Institute of Technology, with a specialization in high-performance computing. His research interests include storage systems, parallel I/O, networks, and high-performance computing.

STEFANO MARKIDIS received the Ph.D. degree from the University of Illinois at Urbana-Champaign. He is currently an Associate Professor in computer science with the KTH Royal Institute of Technology. He has been a Researcher with the Los Alamos National Laboratory, Lawrence Berkeley National Laboratory, and KU Leuven. He was a recipient of two research and development 100 awards, in 2005 and 2017. His research interests include parallel programming models, I/O paradigms, and computational physics.

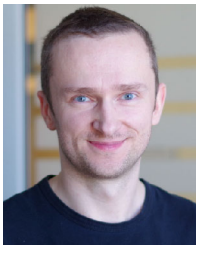
PIERRE GUILLOU received the Graduate and Ph.D. degrees from MINES ParisTech, in 2013 and 2016, respectively. His Ph.D. work revolved around parallel image processing algorithms for embedded accelerators. He is currently a Research Engineer with Sorbonne Université. Since 2019, he has been an active contributor to TTK and the author of many modules created for the VESTEC project.

JULIEN TIERNY is currently the Research Director of the French Center for Scientific Research (CNRS), and he is affiliated with Sorbonne Université, where he teaches computational topology and data analysis and visualization. Prior to his CNRS tenure, he was a Fulbright Scholar with the Scientific Computing and Imaging Institute, The University of Utah. He is the Founder and the Lead Developer of the Topology ToolKit (TTK), an open-source library for topological data analysis and visualization. His research interests include topological methods for data analysis and visualization.

JULES VIDAL received the Engineering degree from ENSTA Paris, in 2018, and the Ph.D. degree from Sorbonne Université, in 2021. He is currently a Postdoctoral Researcher with Sorbonne Université. He is an active contributor to the Topology ToolKit (TTK), an open source library for topological data analysis. His notable contributions to TTK include the efficient and progressive approximation of distances, barycenters, and clusterings of persistence diagrams.

CHARLES GUEUNET is currently a Lead Developer Engineer with the Scientific Visualization Team, Kitware Europe. He joined Kitware, in February 2016. During the first three years with Kitware, he worked on his Ph.D. thesis on "High performance level-set based topological data analysis" in cooperation with Sorbonne Université. During his research work, he became one of the main contributors of the TTK library. After defending his thesis, he joined the Scientific Visualization Team, where he mainly works on maintaining VTK and TTK. His main subjects are high-performance computing, meshing, and discrete geometry.

JOHANNES GÜNTHER received the Ph.D. degree in computer science from Saarland University. He is currently a Principal Software Engineer with Intel, where he is working on high-performance visualization libraries, in particular as a Tech Lead for the OSPRay ray tracing engine. Previously, he was a Senior Researcher and a Software Architect with Dassault Systèmes 3DEXCITE for many years, working on photo-realistic rendering software.



MIROSLAW PAWLOWSKI is currently a Ray Tracing Software Engineer with Intel Technology Poland. His work focuses on developing Intel®oneAPI Rendering Toolkit, mainly Intel®OSPRay rendering engine. His research interest includes the performance optimizations of visualization algorithms.

PIERO POLETTI received the Ph.D. degree in mathematics from the University of Trento. He currently holds a research position with the Bruno Kessler Foundation, Trento, Italy. He is an epidemic modeler. His primary research interests include the investigation of the dynamics of infectious diseases using computational models applied to real-world data with the goal of informing public health decisions, improving the understanding of observed epidemiological patterns, and evaluation of alternative public policies, by identifying risk populations and quantifying the contribution of different factors (e.g., mobility and mixing patterns, demography, and spontaneous behavioral responses to the risk of infection) in shaping the transmission of infectious diseases in humans. His expertise includes the development, simulation, and analysis of compartmental and agent-based models, the adoption of statistical inference and Bayesian approaches to analyze epidemiological records, and the performance of cost-effectiveness analyses. His past research focused on a variety of diseases, including COVID-19, measles, varicella-zoster virus, zika, chikungunya, dengue, respiratory syncytial virus, and influenza.

GIORGIO GUZZETTA received the M.Sc. degree from the University of Pisa, in 2006, and the Ph.D. degree in information and communication technology from the University of Trento, in 2011. He currently holds a tenure track position with the Center for Health Emergencies, Bruno Kessler Foundation, Trento, Italy. He is also a biomedical engineer. His research interests include the modeling of infectious diseases with a focus on public health applications, i.e., risk assessment, outbreak investigation, and the evaluation of the effectiveness of interventions. He is also an Editorial Board Member of *BMC Public Health*.

MATTIA MANICA received the Ph.D. degree in public health and infectious diseases. He is currently a Researcher with Fondazione Bruno Kessler, Italy. He is a mathematical modeler. His past work spans from models of transmission dynamics of vector-borne diseases to the statistical analysis of control interventions, surveillance data, and ecological interaction of vectors. His experience includes both Bayesian and frequents approaches in statistical modeling and classical mathematical models of differential equations. His current research interests include the epidemiology of SARS-CoV-2 transmission and mitigation measures.

AGNESE ZARDINI received the Ph.D. degree in mathematics from the University of Trento, Italy. She is currently a Postdoctoral Researcher with the Bruno Kessler Foundation, Trento, Italy. Her research interests include the development of computational models to investigate the dynamics of mosquito population and infectious diseases.

JEAN-PIERRE CHABOUREAU received the Ph.D. degree from Ecole Polytechnique, Palaiseau, France, and the Habilitation ã Diriger des Recherches from the University of Toulouse, Toulouse, France, in 2006. His Ph.D. work focused on atmospheric water vapor and its retrieval from satellite observations. Since 2010, he has been a Physicist (professor equivalent) with the Observatoire Midi-Pyrénées, University Paul Sabatier Toulouse 3. He has authored or coauthored over 100 peer-reviewed publications. He is the co-leader of the French numerical weather model Meso-NH. In addition to the development of the physics of the models, his interests include clouds, aerosols, deep convection, dynamics, and their interactions.

MIGUEL MENDES received the degree in civil protection, possessing wide know-how in emergency management and in the design and development of emergency management software solutions, and emergency preventive planning. He is currently a Coordinator and the Project Manager of international research and development activities with Tecnosylva. He has been managing successfully the activities of the company in European funded research and development projects of the European 7FP and H2020, such as, ALERT4ALL, PHAROS, HEIMDALL, VESTEC, H2020_Insurance, and FIRE-RES.

ADRIÁN CARDIL received the master's degree in forestry and the Ph.D. degree in forestry and fire science. He is currently a Tecnosylva's International Research and Development Researcher and an Associate Professor with the University of Lleida, Spain. He was a Lecturer in the first European M.Sc. on forest fires (Master FUEGO) with the University of Lleida, Córdoba, and León, Spain, about fire simulation and analysis, and he has been invited to provide training in forest fire related seminars in Italy and Germany. His past work has been already reflected on more than 42 articles in journals of international impact, 18 of these as first author. He has participated with oral presentations and posters in international conferences across Europe and the USA, in international mobility programs in Freiburg, Germany; Sassari, Italy; Montreal, Canada; and participated actively in European Projects (Fire Paradox European Project, New Forests, PHAROS, HEIMDALL, VESTEC, and PYROLIFE) and he has a track on collaborating with foreign researchers.

SANTIAGO MONEDERO received the bachelor's degree in physics and the European Ph.D. degree in applied mathematics, which provides him with a wide perspective of all the aspects related to the challenging task of simulating natural phenomena, and the Ph.D. degree in mathematics. He is currently a Tecnosylva's Chief Modeler and a Main Investigator for environmental related simulation software projects. He has been researching on wind and forest fire simulation for the past 18 years. He has participated in all European projects Tecnosylva has participated or is participating to (e.g., PHAROS, HEIMDALL, VESTEC, OASIS H2020_Insurance, and FP6 PREVIEW).

JOAQUIN RAMIREZ received the Ph.D. degree in forestry engineering. He is currently the CEO of Tecnosylva and a Coordinator and a Lecturer of the first European M.Sc. on forest fires (www.masterfuegoforestal.es). He has technically participated and coordinated the activities of Tecnosylva in the currently ongoing and previous European research and development projects. His research interests include forest fire modeling and geospatial-tools. He is the leading fire scientist, a software architect, and a chief designer of Wildfire Analyst (www.wildfireanalyst.com) and fiResponse (www.firesponse.com) software products.

ANDREAS GERNDT received the degree in computer science from Technical University, Darmstadt, Germany, in 1993, and the Ph.D. degree in computer science from RWTH Aachen University, Germany. He is currently a Professor and the Head of the Department of Software for Space Systems and Interactive Visualization, German Aerospace Center (DLR). In the position of a research scientist, he was also with the Fraunhofer Institute for Computer Graphics (IGD), Germany. Thereafter, he was a software engineer with several companies, with a focus on software engineering and computer graphics. In 1999, he continued his studies in virtual reality and scientific visualization with RWTH Aachen University. After two years of interdisciplinary research activities as a Postdoctoral Fellow with the University of Louisiana, Lafayette, USA, he returned to Germany, in 2008, to work for DLR in the domain of aerospace software research. Since 2019, he has been a Professor in high-performance visualization with the University of Bremen, Germany. His research interests include the interactive visualization of very large scientific datasets, immersive environments, augmented reality, high-performance computing, model-based system engineering, and model-driven software development for space systems.