IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ADAPTIVE INTEGRATED STEERING FRAMEWORK FOR SIMULATION AND VISUALIZATION OF CYCLONE TRACKING ACROSS DIFFERENT APPLICATION CONFIGURATIONS

[1]Author: B Rebecca, Research scholar-CSE department at Sri Satya Sai University of Technology & Medical Sciences

[2]Author: Dr. Pankaj Kawadkar, Professor at CSE department at Sri Satya Sai University of Technology & Medical Sciences

ABSTRACT

Visualization is necessary for further data processing and to assist scientists in comprehending the enormous amount of data generated. Large-scale simulations for essential meteorological applications such as cyclone tracking and earthquake modelling necessitate simultaneous online/"on-the-fly" visualization. Online visualization allows scientists to provide real-time feedback to the simulations, allowing them to better tailor the output to the scientific demands. The main aim of this study is to discuss the adaptive integrated steering framework for simulation and visualization of applications like cyclone tracking across various resource configurations.We developed InSt, an integrated user-driven and automated steering framework for simulations, online remote viewing, and analysis for crucial weather applications, in this research. InSt gives the user control over a variety of application parameters, including the region of interest, simulation resolution, and data visualization frequency. Unlike previous efforts, our framework considers both the steering inputs and the application's criticality, namely the application's minimum progress rate, as well as various resource constraints such as storage space and network bandwidth, to determine the best possible parameter values for simulations and visualization.

***Keywords**: Steering, critical weather application, remote visualization, adaptation, parallel simulation etc.*

1. INTRODUCTION

Impacts of climate change can be felt across all industries and levels of society. Reduced sensitivity to climate change has become a significant concern for the world's poorer countries in recent years. These countries not only lack the resources to deal with climate change, but their economies are also more reliant on climate-sensitive sectors including agriculture, water, and coastal zones. Climate change adaptation remains at the forefront of any sustainable development policy agenda for these countries.

Synoptic and dynamical interpretations of these sensitivities have been noticeably absent from many previous applications of adjoint-based sensitivity analysis to tropical and extratropical cyclone concerns. When a dynamical explanation of sensitivities is given, it is frequently just a

statement that the distribution of sensitivities or singular vectors coincides with a synoptic feature. The mere coincidence of adjoint sensitivities with a synoptic feature does not imply that the feature has dynamical importance. Langland et al. (1995) show that the sensitivity fields calculated in their analysis of an idealized cyclogenesis event in a nonzonal, time-evolving flow were remarkably similar to those calculated for a constant, zonal fundamental condition. While adjoint sensitivities do not provide information about whether certain physical processes occurred within a basic state (control) predicted trajectory, they do provide information about the impact of potential perturbations to the basic state. Adjoint sensitivities have this unique feature, which makes them a potentially effective tool in synoptic case studies.

The progress of numerical weather prediction (NWP) models has substantially improved tropical cyclone (TC) track forecasts during the last few decades. A 72-hour prediction's mean worldwide model track errors are now comparable to those of a 12-hour forecast 25 years ago. For example, in the Western North Pacific, the average 72-hour track forecast error of the UK Met Office's Unified Model (MetUM) has fallen from around 600 km to under 200 km since 1992, while the 48-hour average TC track forecast error is under 100 km. Despite these advances, there are still instances where a TC prediction has significant track inaccuracies. Identifying and comprehending scenarios where a TC's motion is difficult to forecast is critical for developing effective warnings and thereby reducing the storm's potential impact. Furthermore, identifying flaws in the models will assist focus future model development, which should result in better projections.

Adaptation is also a process by which individuals, communities, and countries attempt to adapt with the effects of climate change, such as variability. Adaptation is not a new concept; people have been adapting to changing environments for millennia, including natural long-term climate changes. The concept of factoring future climate risk into policymaking is novel. Although our awareness of climate change and its potential consequences has improved, the provision of practical climate change adaptation assistance has not. Each of these concepts is underpinned by the APF's strong emphasis on flexibility. Users will find a thorough overview of available analytical techniques in the APF, as well as strong advice to employ just those techniques that match their specific needs.

Weather modelling, for example, necessitates high-fidelity simulations including sophisticated numerical models and large-scale calculations that generate a vast amount of data. Visualization is necessary for further data processing and to assist scientists in comprehending the enormous amount of data generated. Large-scale simulations for essential meteorological applications such as cyclone tracking and earthquake modelling necessitate simultaneous online/"on-the-fly" visualization. Online visualization allows scientists to provide real-time feedback to the simulations, allowing them to better tailor the output to the scientific demands. Geographically dispersed climate experts can use remote visualization to communicate essential data, collaborate on analyses, and provide joint counsel on critical weather events. Thus, a vast climate science community may use remote viewing and feedback management via computational steering to analyses large-scale scientific simulations.

Because some meteorological events might be connected with these data, it is necessary to present the data alongside its geographical context (e.g., time-

independent data such as terrain, cities, and water bodies) for visual interpretation of weather simulations. In addition, integrating relevant observational data is required in order to visually compare this data with simulation results and detect links and contradictions. As a result, meteorologists must explore and analyses data sets from multiple sources (multimodal) that are extremely complicated. These data sets contain spatiotemporal data as well as a variety of factors (multivariate). Furthermore, data sets containing numerous models runs (multirun) must be evaluated. Scientific visualization approaches for multidimensional data must be created in order to explore, evaluate, and present such data.

We develop a workflow and an application with experts in meteorology and visualization that can be used collaboratively to verify, falsify, and generate hypotheses, study phenomena and their evolution, discover inconsistencies and unexpected features in the evolution and representation of atmospheric processes. For time-dependent meteorological data, we combine common 3D visualization approaches and build a visualization programme with a graphical user interface (GUI) that assists users in evaluating these data sets. To address this problem, we propose a solution that combines the flexibility and power of a visualization tool with the interactivity and presentation capabilities of computer gaming engines.

In today's market, active steering is a cutting-edge technology that electronically adjusts the steering wheel angle. Variable steering ratio and an interface to support the vehicle dynamic control system are among the features of Active Steering. Additionally, this steering assist technology is used to lessen wind gusts, wind force disturbances, and increase stability while driving on slick roadways. The system is activated when it detects abnormal steering behaviour, which in this example is quick left and right turning movements. The light emission diode signal is detected using a light sensor.

2. LITERATURE REVIEW

John Ashcroft, et al (2021) - Typhoons Haiyan (2013) and Hagupit (2014) are two tropical cyclones (TCs) whose predictabilities diverged substantially despite similarities in track and intensity. Both TCs made landfall over the Philippines after taking a similar path across the Pacific and reaching intensities of more than 60 ms1. The course of Hagupit exhibited a lot of uncertainty in operational global ensemble forecasts, but the ensemble spread for Haiyan was much smaller. For both storms, 5-day global ensemble forecasts were created using the Met Office's Unified Model. The spread of tracks in the forecasts made for Hagupit is higher than in the forecasts produced for Haiyan, which is consistent with the operational forecasts produced at the time of the storms. For an accurate forecast, it's critical to accurately reflect both the steering flow and Hagupit's position.

Ayisha Siddiqua L and Senthil kumar N C (2019) - In existing systems, it is possible that the data is inaccurate and that adequate data mining techniques are not being applied, which adds to the complexity. We humans are prone to making errors when anticipating weather conditions, which can lead to loss of life and property. To avoid this, we employ data mining algorithms to estimate rainfall based on meteorological factors such as maximum temperature, minimum temperature, wind speed, humidity, pressure, dew point, cloud, sunshine, and wind direction. However, by employing adequate dataset methods and employing the appropriate criteria, we can attain reliable rainfall prediction findings. As a result, we use the Decision Tree technique with the Gini Index to accurately estimate precipitation, and it is fully based on historical data.

Siddharth S. Bhatkande and Roopa G. Hubballi (2016) - "Weather forecasting is a critical application in meteorology, and it has long been one of the world's most scientifically and technologically difficult problems." We look into how data mining techniques can be used to forecast qualities like maximum and minimum temperatures. This was done using Decision Tree algorithms and meteorological data collected from several cities between 2012 and 2015. Complex meteorological phenomena make it difficult to anticipate the weather. Many elements of weather events, such as maximum temperature, minimum temperature, humidity, and wind speed, are

impossible to count and measure. We use the Decision Tree Algorithm to delete unwanted data from available datasets. In general, maximum and minimum temperatures are the most important factors in weather forecasting. We anticipate a full cold, full hot, or full snowfall based on the percentage of these parameters. This work presents a decision tree-based model to anticipate meteorological phenomena such as extreme cold, extreme heat, and snowfall, which can be life-saving information.

Carolin Helbig, et al (2015) - Meteorologists build and employ models with increasing spatial and temporal resolution to achieve more accurate simulations. Due to the expanding volumes and multidimensional nature of the data, interpreting, comparing, and displaying the generated simulations becomes increasingly difficult. A complicated database is created by combining several data sources, numerous variables, and multiple simulations. Although there is software available for the visualization of meteorological data, none of it meets all of the domain-specific requirements: support for quasi-standard data formats and different grid types, standard visualization techniques for scalar and vector data, visualization of the context (e.g., topography) and other static data, support for multiple presentation devices used in modern sciences (e.g., virtual reality), and a user-friendly interface.

Brett T. Hoover and Michael C. Morgan (2010) - The advection of a tropical cyclone (TC) vortex by a "environmental wind" is typically interpreted as the steering of a TC vortex. The environmental steering wind vector has previously been described as the horizontal and vertical averaging of horizontal winds in a box centred on the TC in previous investigations. To generate adjoint-derived sensitivities of TC zonal and meridional steering, the components of this environmental steering have been proposed as response functions. A two-dimensional barotropic model and its adjoint for a 24-hour forecast are used to examine the applicability of these response functions in adjoint sensitivity analyses of TC steering. Because perturbations to the model initial conditions that affect the TC's final-time location also modify the response functions in ways that have nothing to do with the TC steering during model verification, these response functions do not produce sensitivity to TC steering.

1. InSt STEERING FRAMEWORK

We created InSt, an integrated steering framework that conducts both automatic and user-driven steering. It consists of the following components, as shown in Figure 1: an application manager that determines the application configuration for weather simulations based on resource characteristics and user input, a simulation process that simulates with different application configurations, a visualization process for frame visualization, frame sender and receiver daemons that deal with frame transfer from simulation to visualization sites, simdaemon and visdaemon of The key components are described in detail in the subsections that follow.

- ✓ **User Interface, SimDaemon and VisDaemon -** Figure 1 shows how the user interacts with the user interface at the visualisation site. The user interface allows the user to define nest location, simulation resolution, and output interval and simulation progress rate bounds. The application manager receives the user's input values via the VisDaemon and SimDaemon. Through the visualisation process, the VisDaemon gets user input from the user interface and transfers it to the SimDaemon. This user input is specified by the SimDaemon to the programme management. The daemons communicate the manager's reaction to the user through the user interface.
- ✓ **Application Manager -**The application manager is the most important part of InSt, serving as a link between algorithmic and user-driven steering. The manager checks the available network bandwidth and free storage space on a regular basis. Given the resource limitations and simulation settings, it regularly calls the decision algorithm to determine the number of processors for simulation and the frequency of output to be generated by the simulation. The application manager asynchronously collects user inputs, including the upper bound for output frequency and simulation resolution, allowing user-driven steering. If conducting the simulation with the user inputs is not practicable, the manager tells the users of alternate possibilities and, if viable, activates the decision algorithm using the user inputs and resource parameters. The manager saves the decision algorithm's output in the application configuration file and begins, stops, and restarts the simulation using the parameters.

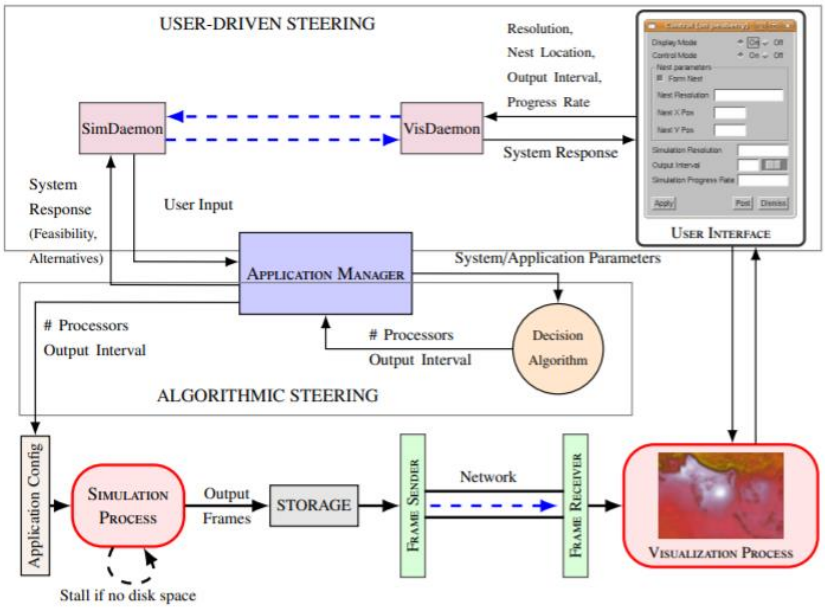


Figure 1: InSt: Integrated Steering Framework – Existing model

- ✓ **Simulation Process** -The weather application that mimics weather events is the simulation process. It reads the simulation parameters from the application configuration file on a regular basis. The simulation process stops and resumes execution with the new parameters if the parameters in the configuration file differ from the values used for current execution. It also uses sleep to halt operation if the available free disc space falls below a certain level. It examines the disc space on a regular basis and only continues processing when the disc space is available again.
- ✓ **Decision Algorithm** -The decision algorithm calculates the number of processors and the frequency with which weather data is output for a given simulation resolution, as well as the network bandwidth between simulation and visualization sites, the available free disc space at the simulation site, and the user-specified minimum simulation progress rate. The programme also takes into account execution times for various processor counts and simulation resolutions. It also takes into account the lower constraint for output frequency or the higher bound for the gap between outputs, known as the upper output interval. This upper bound is the minimal frequency with which a climate scientist wants to visualize weather events.

The decision algorithm's goal is to increase the rate of simulations and provide continuous display with the highest temporal resolution possible. These goals, however, are incompatible. Increasing the frequency of simulation output can reduce the rate of simulations due to an increase in I/O, as well as result in rapid storage consumption. Unlike standard scheduling algorithms that aim to reduce execution durations, our system may need to "slow down" simulations on occasion, as faster simulations can result in higher storage consumption if network bandwidth is limited. As a result, we can consider our challenge to be an optimization problem in which we try to optimise the simulation rate while staying within the restrictions of continuous visualisation, acceptable output frequency, MPR of simulations, I/O bandwidth, disc space, and network speed. We used a linear programming model to solve this problem, with the goal of achieving the number of processors and frequency of output while meeting the limitations of minimal stalling at the visualisation end and disc capacity for storing the simulation output. For addressing the criticality of the application and assuring quality of service to the climate scientist for online display, we've introduced a rate constraint using the MPR of simulations. The constraints and objective function are described further down. The goal is to increase simulation throughput or reduce t, where t is the time it takes to solve a time step. The parameters that were utilised to create the formula are listed below. The choice variables are $S, F, T\ D$, and t.

t: Time to solve one simulation time step	$\mathcal{S}$: Number of frames solved in an interval
$\mathcal{F}$: Number of frames output in an interval	$\mathcal{T}$: Number of frames transferred in an interval
O: Size of one frame output in one time step	$\mathcal{D}$: Total remaining free disk space
$\mathcal{T}_{IO}$: Time to output one time step	b: Network bandwidth

The ratio of the time simulated by the application (simulation time) to the time taken by the application for the simulation (wall-clock time) is presented (wall-clock time). For crucial applications such as cyclone tracking, where scientists require advance information to provide timely recommendations to decision makers, this ratio must be greater than one. The simulation rate is determined by the output frequency and the computation speed. The number of I/O writes to the disc will increase as the output frequency increases, and the simulation rate will decrease. In addition, the lower the I/O bandwidth, the greater the influence of high output frequency on simulation rate. In most circumstances, scientists will want to define an MPR (minimum acceptable ratio) restriction for this ratio. This constraint is stated in Equation (1). The integration time step associated with a simulation resolution is denoted by ts. In one time step, it is the amount of time simulated. If S frames are solved and F frames are created in interval I, simulation time is ts S, and wall-clock time is the sum of solution and I/O time i.e. $t \cdot S + T_{10} \cdot F$ as demonstrated by Equation (2). Equation (3) can be rearranged to produce the rate constraint equation.

$$\frac{Simulation\ time}{Wall-clock\ time} \geq MPR \qquad (1)$$

$$\frac{(ts.S)}{(t.S + T_{\mathrm{IO}}.F)} \geq MPR \qquad (2)$$

$$\frac{ts}{(t + T_{\mathrm{IO}}.F/S)} \geq MPR \qquad (3)$$

- ✓ **Other Constraints**: Because the next set of F frames should be available for transfer by the time the current T frames are transferred, the time it takes to solve S frames and produce F frames in an interval I should be shorter than the time it takes to transfer T frames in that interval. The time limitation is formed as a result of this. The rate of input to the disc is affected by I/O bandwidth, computing speed, and output size, whereas the rate of output from the disc is affected by network bandwidth. We create the disc constraint using them.

$$t \geq T_{LB} \quad (4)$$

$$OI_{LB} \leq OI \leq OI_{UB} \quad (5)$$

- ✓ **Bounds**: t has a lower bound based on the total number of processors. T_{LB}, according to Equation (4). Based on the minimal frequency of visualization of weather occurrences defined by climate scientists, we can select an upper constraint OIUB for the output interval. Based on the simulation application's restrictions, the output interval has a lower bound OILB. In Equation, these bounds are defined (5). OIUB can be selected or directed by the user in InSt.

By changing variables, we can linearize non-linear constraint equations. We used GLPK to address our optimization challenge (GNU Linear Programming Kit). The number of processors is then calculated using t, and the output interval is calculated using F/S and ts. During the simulation run, this decision

process is called on a regular basis. This algorithm outputs t and OI to the application configuration file, which is used to reschedule the simulations with the new configuration, given the inputs D, TIO, b, and O.

4. PROPOSED METHODOLOGY

4.1 Reconciling User-Driven And Algorithmic Steering

Initially, the application manager sets the simulation resolution and MPR to default levels. The manager then uses a judgement procedure based on resource restrictions to decide the frequency of output for the particular simulation resolution. These settings are used to start the simulation. These simulation parameters can be changed during execution by the user at the visualisation site using the user interface. The user can additionally designate a place for the development of a nest or a sub-region in the domain for finer resolutions using the interface. The simulation procedure resets and continues with a fresh configuration involving the nest and the new resolution when the user requests nest placement. When a user specifies an output interval and/or MPR value, the InSt framework examines the application's criticality, checks the feasibility of simulating with these inputs, and suggests alternate values to ensure continuous simulation and visualisation progress. When a user specifies an output interval, the framework determines whether the specified interval can be used without violating Equation (3)'s rate constraint, that is, whether the simulation can generate output with the specified interval while maintaining a simulation rate greater than the application manager's current MPR. Equation (6), which is calculated using Equation (3) and the equation F/S = ts/OI stated in, describes the link between OI and MPR.

$$MPR \leq \frac{ts}{[t + T_{IO}.(ts/OI)]} \quad (6)$$

The direct relationship between OI and MPR may readily be recognized. Let's say the resolution is 15 kilometres, the integration time-step is 45 seconds, TIO is 43 seconds, and the time interval is 3.95 seconds. Now, if the user wants an OI of 180 seconds, MPR will be less than or equal to 3.06 based on Equation (6). However, if the current MPR is 5, the system will plainly be unable to fulfil his request for an OI of 180 seconds. As a result, InSt must overrule the user's requested OI value in such circumstances. Equation (6) also shows that if OI is too low, the simulation rate will be reduced as well. As a result, InSt uses the current value of MPR used by the application manager to derive the lower bound of OI from Equation (6) in order to simulate and visualize at a constant rate. IS does not include the request if the user-specified OI is less than this lower constraint. In this situation, InSt displays the lowest possible OI to the user. If the user-specified OI exceeds the lower bound, the upper bound for OI in the decision algorithm is set to this number.

If a user specifies an MPR value, the application manager tries to replace the current MPR value used in the decision algorithm with the value supplied by the user. However, it first determines whether the user-specified MPR, uMPR, is viable for the present simulation resolution by computing an upper bound, MPR_{max}, and comparing uMPR to MPR_{max}. In Equation (6), we replace OI with ∞ and t with its lower bound TLB to get MPR_{max} as the ratio of ts and TLB, which is the feasible upper bound. If uMPR is greater than MPR_{max} for the current resolution, InSt proactively searches for a coarser resolution and calculates MPR_{max} for the coarser resolution to ensure that uMPR is feasible. The programme manager then gives the user the option of using a coarser resolution using the uMPR. The WRF simulation is restarted with the nest if the user desires to place one in the parent domain. InSt tests for the feasibility of the combination if the user specifies both uMPR and uOI. If the uMPR is possible, InSt includes it; otherwise, the framework checks for uMPR feasibility at a coarser resolution, because the rate of simulation increases as the resolution is coarser. If the user merely specifies uOI, InSt determines whether uOI fulfils the current MPR. If possible, the simulation is updated; if not, the user is notified. As a result, InSt seeks to strike a compromise between the decision algorithm's algorithmic steering and the user-driven steering settings, taking into account the application's criticality (expressed by MPR).

5. RESULTS AND DISCUSSIONS

5.1 Resource Configuration

All of our studies were carried out on a graphics workstation at the Indian Institute of Science (IISc) with a dual quad-core Intel R Xeon R E5405 processor and an NVIDIA GeForce 8800 GTX graphics card. For faster visualization, we employed It's hardware acceleration capability. The simulations were run on two different sites, yielding two different remote visualization and computational steering settings: intra-country and inter-country steering. The simulations were run on the 3.16 GHz quad-core Intel R Xeon R X5460 320-core cluster, gg-blr, at the Centre for Development of Advanced Computing (C-DAC), Bangalore, India, in the intra-country configuration. For this setup, we used the Gigabit interconnect. The National Knowledge Network was used to carry data between the simulation and visualization sites, with a maximum capacity of 1 Gbps and an average accessible bandwidth of 40 Mbps. For this configuration, we used a maximum of 96 cores and 150 GB of storage space. The WRF simulations were run on a dual-socket quad-core Intel 64 (Clovertown) PowerEdge 1955 9600-core cluster, Abe, at the National Center for Supercomputing Applications (NCSA), Illinois, USA, in the inter-country configuration. For this simulation, we employed the Infiniband link. A maximum of 128 cores and 700 GB of storage space were employed. This configuration has an average available bandwidth of 8 Mbps.

5.2 Weather Model and Cyclone Tracking

For our research, we employed the WRF (Weather Research and Forecasting Model) mesoscale numerical weather forecast model to simulate weather events. In WRF, the forecasted region is referred to as a domain. In WRF simulations, there is a parent domain that can contain offspring domains, which are referred to as nests. WRF allows you to use nesting to run simulations at a finer level in specific areas of interest. WRF generates NetCDF files as its output. The output of a number of simulation time steps is contained in each NetCDF file. We used InSt to follow cyclones on a big scale and over great distances, which is a key weather application. In our tests, we used InSt to track a tropical storm named Aila that formed in the Northern Indian Ocean in May 2009. We used WRF's nesting function to follow the cyclone's lowest pressure zone, or eye, and run finer level simulations in the area of interest. The nesting ratio, which is the ratio of the nest's resolution to the parent domain's, was set to 1:3. We ran simulations for an area of roughly 32×10^6 sq km from 60^0E to 120^0E and 10^0S to 40^0N over a two-day period for the intra-country trials. We ran simulations over a greater area or domain, from 30^0E to 150^0E and 10^0S to 40^0N, and for a longer length of time, 3 days and 18 hours, for the inter-country trials. The CISL Research Data Archive provided us with 6-hourly 1-degree FNL analysis GRIB meteorological input data for our model domain.

5.3 Implementation of Framework

Our work necessitated just minor changes to the WRF application. The primary change is that when the application configuration file provides a different number of processors and output interval than the present setup, WRF will not reschedule on those processors. The default upper bound for output frequency was set to 30 simulated minutes, while the lower bound was set to 3 simulated minutes for our tests. During steering, however, the user can alter these parameters. Sample WRF runs, each with a simulation period of 1 hour, were executed for different discrete numbers of processors spanning the available processor space and using curve fitting tools to interpolate for other numbers of processors to obtain the simulation rates that are used by our decision algorithm. In the gg-blr cluster, these WRF profiling runs were conducted on 16, 24, 32, 48, 56, 64, 80, and 96 processors, whereas in the abe cluster, they were run on 32, 48, 64, 80, 96, 112, and 128 processors. We used WRF's split NetCDF technique for quicker I/O, where each processor outputs its own data. At the visualization site, we created a programmed to reconcile these divided NetCDF files. We've also created a VisIt plug-in that reads WRF NetCDF output files directly, removing the cost of post-processing before data analysis. After arriving at the visualization site, we adjusted VisIt to render automatically as and when these WRF NetCDF files are merged. We used volume rendering, vector plots with oriented glyphs, pseudocolor, and contour plots to visualize the results. We used Qt to create a GUI for the steering interface in VisIt. Figure 1 shows a screenshot of the user interface. Using the UNIX command df, the application manager checks the

available disc space on a regular basis. The time it takes to deliver a 1 GB message across the network yields the average observed bandwidth between the simulation and visualization locations.

Every 1.5 hours, the application management additionally employs our decision algorithm. This frequency was enough for our test conditions, as the network bandwidth did not fluctuate much. The decision method will have to be used more frequently in highly dynamic contexts. Despite the fact that the threshold values employed in IS are unique to our experience, only minimal changes to the WRF application were required for our work. The primary change is that if the application configuration file provides a different number of processors and output interval than the present setup, WRF will no longer reschedule on those processors. For our experiments, the maximum bound of the default output frequency was set to 30 simulated minutes, and the lowest bound was set to 3 simulated minutes. The user can, however, change these variables while steering. Sample WRF runs with a 1-hour simulation period were done for different discrete numbers of processors across the available processor space and interpolated for other numbers of processors using curve fitting techniques to produce the simulation rates used by our decision algorithm. These WRF profiling runs were carried out on 16, 24, 32, 48, 56, 64, 80, and 96 processors in the gg-blr cluster, and on 32, 48, 64, 80, 96, 112, and 128 processors in the abe cluster. For quicker I/O, we employed WRF's split NetCDF technique, which allows each processor to produce its own data. We designed a method to fuse the divided NetCDF files at the visualization site. We've also developed a VisIt plug-in that reads WRF NetCDF output files directly, which saves time and money when it comes to data processing. We customized VisIt to render these WRF NetCDF files automatically once they arrived at the visualization location. The results were visualized using volume rendering, vector plots with oriented glyphs, pseudocolor plots, and contour plots. For the steering interface, we utilized Qt to construct a GUI within VisIt. A screenshot of the UI is shown in Figure 1. The application manager maintains track of how much disc space is available using the UNIX command df. The average observed bandwidth between the simulation and visualization sites is calculated using the time it takes to deliver a 1 GB message across the network.

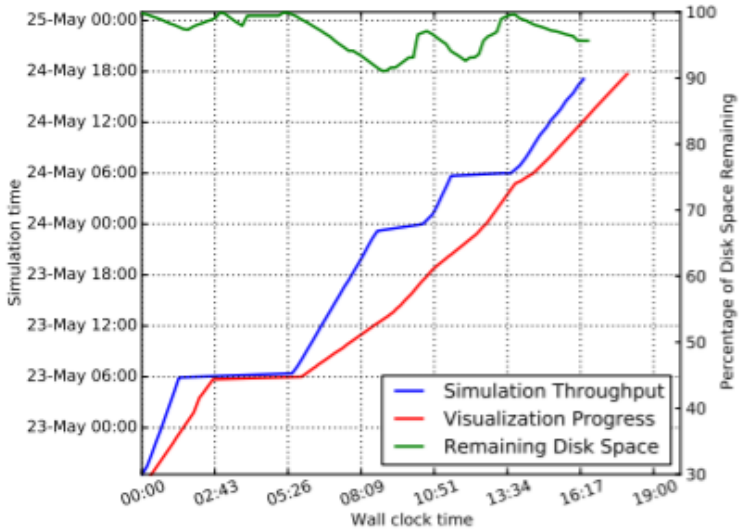
Our decision method is used by the application management every 1.5 hours. Because the network bandwidth did not fluctuate considerably during our tests, this frequency was sufficient. In highly dynamic environments, the decision procedure will have to be employed more frequently. Although the threshold values used in IS are specific to our test settings and WRF simulations, the fundamental concepts of our steering architecture are universal and can be used to a variety of applications. Situations and WRF simulations, our steering framework's main concepts are generic and transferable to various applications.

5.4 Automatic Tuning Results

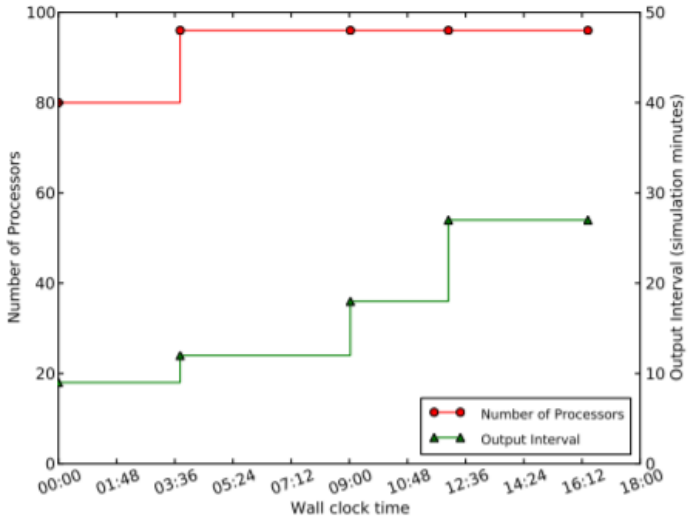
In the absence of user inputs, we first demonstrate the efficiency of the decision process and the automatic tweaking of parameters. InSt automatically finds the WRF nest position and modifies the resolution of simulations based on pressure values in the weather data for the experiments in this area. When the pressure falls below 995 hPa, InSt creates a nest in the parent domain, centred on the lowest pressure point. A configuration file is also used, which provides multiple simulation resolutions for different pressure gradients or cyclone intensity. Climate scientists can specify this by using coarser resolutions in the early phases of cyclone formation and finer resolutions as the cyclone intensifies. InSt modifies the resolution numerous times when the cyclone develops, i.e., the pressure drops even lower, in order to get a better simulation result from the model.

The simulation's initial resolution was set at 24 km for the intra-country configuration. When the lowest pressure falls below 994 hPa, the resolution is altered to 21 km, 18 km for pressures less than 991 hPa, 15 km for pressures less than 989 hPa, and 12 km for pressures less than 987 hPa. The results for the intra-country experimental setting are shown in Figure 2(a). The graph depicts three curves: a simulation curve (blue) that plots the simulated time against wall clock time, a visualization curve (red) that plots the times at which the related output was visualized, and a disc availability curve (green) that depicts the available storage space. The left y-axis indicates the relevant simulation times of the frames for the simulation and visualization curves. The right y-axis of the disc availability curve represents the proportion of remaining disc space. The latency between visualization and simulation is small in Figure 2(a),

resulting in a genuinely online visualization. The reason for this is due to the high network bandwidth. However, because of differences in network bandwidth and simulation rates at different times of execution, the lag is not constant. WRF requires input data at that resolution when it restarts at a finer resolution. These areas are represented by the curve's flat areas. The extended flat sections are owing to the gg-blr cluster's unusually low I/O bandwidth. Because of the continual transmission of frames from the simulation site to the visualization site, even during these restart occurrences in simulation, these regions also correlate to an increase in available disc space. Because of the high network bandwidth, disc space was freed at a rapid rate, the remaining disc space is always above 90%.

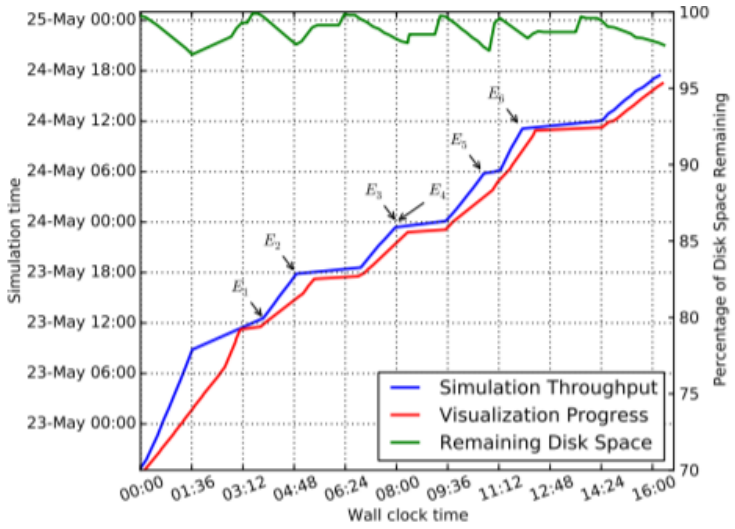


(a) Progress of Simulation (blue) and Visualization (red), as well as Disk Consumption (green). Due to the high network bandwidth, frames are shown with minimal time lag.

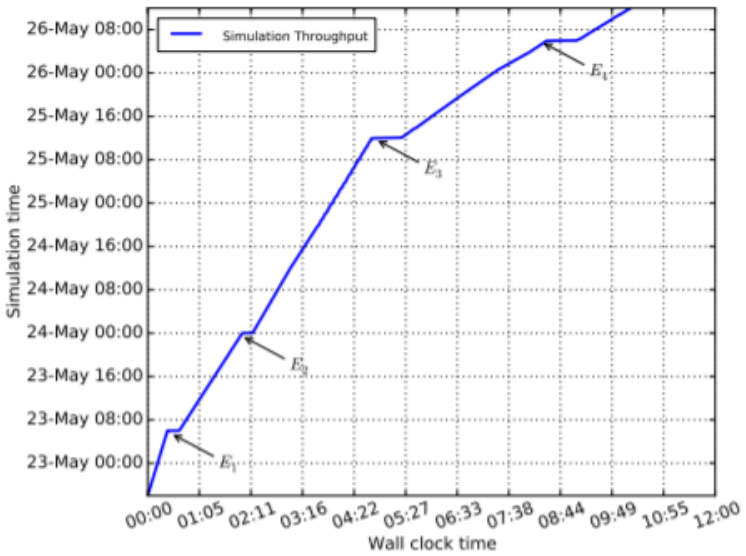


(b) The framework's adaptivity, as measured by the number of processors (red, left y-axis) and output interval (green, right y-axis).

Figure 2: Progress in simulation and visualization, disc consumption, and adaptivity for intra-country algorithmic steering



(a) For an intra-country setup with computational steering, simulation progress (blue), visualization progress (red), and disc use (green) are shown



(b) Inter-country setup with computational steering simulation progress. The simulation throughput is affected by both algorithmic and user-driven steering events (E_1–E_4).

Figure 3: In intra-country and inter-country arrangements, the effects of algorithmic and user-driven steering events

Figure 2(b) depicts the number of processors and output interval for the intra-country configuration simulations automatically selected by InSt at various phases of execution. IS starts with a value of 9 for the output interval and 80 for the number of processors, respectively. When one or more parameters in the constraint equations change, the number of processors and output interval vary a few times during execution. When the resolution is reduced from 18 km to 15 km, the time it takes to solve a time step, the size of the output data per time step, and the I/O time all change. As a result, the decision algorithm re-evaluates the optimal number of processors and output interval for the present parameters. The output size and, as a result, the I/O time grows as the resolution gets finer. MPR was chosen as the number 5 for this experiment. To fulfil the MPR and avoid disc overflow owing to frequent I/O, the decision algorithm raises the output

interval. As a result, based on the application and resource setups as well as simulation rates, InSt algorithmically directs and alters the execution and application parameters.

5.5 Computational Steering Results

In this part, we show how InSt supports user-driven steering, including the various steering capabilities supplied to the user, the framework's feedback mechanisms, and the reconciliation of user-driven and algorithmic steering. The simulations are begun at a certain resolution for these tests. However, unlike the automatic tuning experiments, only the user has control over where the nest is placed and how the simulation resolutions are changed.

5.5.1 Intra-country Steering

The steering results for the intra-country experimental setting employing the gg-blr cluster are shown in Figure 3(a). The graph also depicts the various steering events that the user has specified. The simulation was initially run with a resolution of 24 km and an MPR of 5. Steering events were triggered by user input at various stages of the simulation and visualization. The steering events, as well as the system response and consequence of these events, are listed below.

- E_1: After 3.8 hours of execution, this event happens. In this case, the user decides to create a nest based on the visualization output and also provides the location of the nest. This causes the decision algorithm to recalculate the best number of processors and output interval based on the various system and application parameter values. The construction of a nest reduces the simulation rate, and as a result, the slope following the event lowers.
- E_2: After around 4.8 hours of execution, this event happens. In this case, the user chooses to refine the simulation to an 18-kilometer resolution. WRF must preprocess incoming data at a finer resolution in order to start at that resolution. As a result, following E_2, we see the flat region, which corresponds to the preprocessing time. This time is determined by the system's I/O bandwidth. The framework then begins with an 18-kilometer resolution specified by the user.
- E_3, E_4: After around 8 hours of execution, this event occurs. The user asks an 8-simulation rate at E_3, but the system refuses due to infeasibility. With an output interval of 30 and a resolution of 18 km, the maximum rate feasible is 6. Equation can be used to calculate this (6). As mentioned in Section 4, the system next tries to locate a coarser resolution at which the user's requested rate is achievable. In this situation, the system offers the option of increasing the resolution to 21 kilometres. As a result, the system strives to meet one of the user's requirements while compromising another depending on feasibility analysis. The user can then choose to favour resolution above rate, or vice versa. The user requests a resolution of 21 km, as advised by the system, at event E_4. IS takes a proactive approach to user-driven computational steering, as these occurrences indicate. While attempting to steer simulations based on user inputs, it also analyses the impact of user inputs on the criticality of the application, namely, the MPR desired for simulations, and "advises" the user about possible violations of the "quality of service" as a result of his inputs, as well as providing suitable alternate options. As a result, InSt takes an efficient reconciliation approach to guiding executions. Unlike previous work that has primarily focused on user-driven steering, this reconciliation of user inputs and application criticality requirements is critical for vital weather applications such as cyclone tracking.
- E_5: After approximately 10.6 hours of operation, this event happens. The user decides to raise the output interval to 21 in this case. The slope of the simulation curve increases slightly, indicating an increase in the simulation rate. Because the output interval is extended, the simulation process spends less time writing files to disc, the simulation rate increases.
- E_6: After approximately 11.8 hours of execution, this event happens. The user can choose to fine-tune the resolution for better visualization. Due to the fact that finer

resolution requires more time to solve a time step, the slope of the simulation curve after E_6 is smaller than before.

The available disc space is always greater than 95%, as shown in Figure 3(a). This is due to the decision algorithm increasing the number of processors to the correct value, ensuring that disc space is not a problem even when user inputs are provided. At moments corresponding to events E1 through E6, fluctuations in the disc curve can be detected. Because the transfer rate remains constant when there is essentially no input to the disc during this interval, an increase in disc space can be visible after the events and before restarting WRF.

5.5.2 Inter-country Steering Results

Inter-country steering was also done from the visualization location in India to the simulation site in the United States. Figure 3(b) depicts the inter-country findings achieved using NCSA's Abe cluster in the United States for simulations and the visualization engine at IISc in India. Initially, the simulations were run with an output interval of 30 simulated minutes, a resolution of 18 km, and an MPR of 3 on 96 processors. The algorithmic events (E_1 and E_4) as well as user-driven steering events (E_2 and E_3) that happened during the 11-hour execution are listed below. In terms of simulation throughput and visualization progress, the response to these occurrences is identical to the intra-country experiment.

- E_1: After 30 minutes of execution, this event happens. The decision algorithm calculates the number of processors to be 80 in this case. The change from 96 to 80 is due to the high simulation rate at coarser resolution consuming a lot of disc space quickly.
- E_2: After 2 hours of execution, this event happens. The user requests that the output interval be changed to 60 minutes.
- E_3: After 5 hours of execution, this event happens. In this case, the customer requests that the resolution be reduced from 18 km to 12 km in order to improve simulation results.
- E_4: After 8 hours of operation, this event happens as the simulation rate slows due to finer resolution. In reaction to this event, the decision algorithm increases the number of processors from 80 to 112 while maintaining the minimum simulation pace.

6. CONCLUSION

For efficient monitoring and timely study of crucial weather occurrences, high-performance simulations, effective "on-the-fly" remote visualizations, and user-driven computational steering of simulations based on feedback to focus on important scientific phenomena are required. We explain our integrated steering framework, InSt, in this paper, which combines user-driven steering with automatic tuning of application parameters based on resource restrictions and the application's criticality demands to select the final settings for simulations. InSt assesses the impact of user inputs on the application's criticality in advance, notifies the user of any violations, guides the user through alternative possibilities, and arrives at the final settings. With trials involving intra- and inter-country steering, we proved the algorithmic and steering components of InSt. The results of these tests show how the framework ensures a low simulation rate, continuous display, and a balance between algorithmic and user-driven steering.

7. FUTURE WORK

We intend to look into research difficulties linked to steering across very sluggish networks, comparable to our inter-country setup, in the future. We intend to expand InSt to address the issues of multi-user steering in grid environments and simultaneous steering of several simulations. We also plan to use our ideas in other crucial applications and create a more generic framework.

REFERENCES

1. John Ashcroft, et al (2021). The impact of weak environmental steering flow on tropical cyclone track predictability. Q J R Meteorol Soc. 2021;147:4122–4142.
2. Ayisha Siddiqua L and Senthil kumar N C (2019) - Heavy Rainfall Prediction using Gini Index in Decision Tree. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8 Issue-4, November 2019
3. Siddharth S. Bhatkande and Roopa G. Hubballi (2016) - Weather Prediction Based

on Decision Tree Algorithm Using Data Mining Techniques. International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

4. Helbig C, Bilke L, Bauer H-S, Böttinger M, Kolditz O (2015) MEVA - An Interactive Visualization Application for Validation of Multifaceted Meteorological Data with Multiple 3D Devices. PLoS ONE 10(4): e0123811. https://doi.org/10.1371/journal.pone.0123811
5. Brett T. Hoover and Michael C. Morgan (2010) -Validation of a Tropical Cyclone Steering Response Function with a Barotropic Adjoint Model. Volume 67: Issue 6
6. C. Tapus, I.-H. Chung, J. Hollingsworth, Active Harmony: Towards Automated Performance Tuning, in: SC '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing, 2002.
7. P. Malakar, V. Natarajan, S. Vadhiyar, An Adaptive Framework for Simulation and Online Remote Visualization of Critical Climate Applications in Resource-constrained Environments, in: SC '10: Proceedings of the 2010 ACM/IEEE conference on Supercomputing.
8. J. Brooke, T. Eickermann, U. Woessner, Application Steering in a Collaborative Environment, in: SC '03: Proceedings of the 2003 ACM/IEEE conference on Supercomputing.
9. W. Gu, G. Eisenhauer, E. Kraemer, K. Schwan, J. Stasko, J. Vetter, N. Mallavarupu, Falcon: on-line monitoring and steering of large-scale parallel programs, in: Proceedings of the Fifth Symposium on the Frontiers of Massively Parallel Computation, 1995, pp. 422–429.
10. P. Malakar, V. Natarajan, S. Vadhiyar, A framework for online visualization and simulation of critical weather applications, Tech. rep., Department of Computer Science and Automation, Indian Institute of Science, http://www.csa.iisc.ernet.in/TR/2011/1.
11. GNU Linear Programming Kit, http://www.gnu.org/software/glpk.
12. National Knowledge Network, Department of IT, Government of India, http://www.mit.gov.in/content/national-knowledge-network.
13. J. Michalakes, et al., The Weather Reseach and Forecast Model: Software Architecture and Performance, in: Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology, 2004.
14. J. Michalakes, et al., WRF Nature Run, in: SC '07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing.
15. R. Rew, G. Davis, The Unidata netCDF: Software for Scientific Data Access, in: 6th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, California, American Meteorology Society, 1990.
16. J. A. Insley, M. E. Papka, S. Dong, G. Karniadakis, N. T. Karonis, Runtime Visualization of the Human Arterial Tree, in: IEEE Transactions on Visualization and Computer Graphics, 2007
17. H. Wright, R. H. Crompton, S. Kharche, P. Wenisch, Steering and visualization: Enabling technologies for computational science, in: Future Generation Computer Systems, Vol. 26, 2010, pp. 506–513
18. Opprecht, Wanda & Wanda,. (2014). Conceptual framework for the steering of Information Systems Evolution.
19. Liang, Tong & Beibei, Zhang. (2010). Study on Experiment Platform of Integrated Steering and Braking Control for Vehicle Active Safety System. 2nd International Conference on Information Engineering and Computer Science - Proceedings, ICIECS 2010. 1-4. 10.1109/ICIECS.2010.5678153.
20. Kolmasiak, C. & Budzik, R. & Pardela, W.. (2005). Elements of steering and quality management in integrated computer system of metallurgical enterprise. 44. 63-67.