An On-Demand Weather Avoidance System for Small Aircraft Flight Path Routing ⋆

Eric Lyons[1][0000−0002−9692−0076], David Westbrook[1], Andrew Grote[1], George Papadimitriou[2][0000−0001−9384−5034], Komal Thareja[3][0000−0002−8113−6967], Cong Wang[3][0000−0001−6429−8799], Michael Zink[1], Ewa Deelman[2][0000−0001−5106−503X], Anirban Mandal[3][0000−0001−5145−8618], and Paul Ruth[3][0000−0003−1744−847X]

[1] University of Massachusetts Amherst, Amherst, MA 01003, USA
elyons@engin.umass.edu
[2] Information Sciences Institute, University of Southern California, Marina Del Rey, CA 90292, USA
[3] Renaissance Computing Institute (RENCI), University of North Carolina at Chapel Hill, Chapel Hill, NC 27517, USA

Abstract. Convective weather events pose a challenge to the burgeoning low altitude aviation industry. Small aircraft are sensitive to winds and precipitation, but the uncertainty associated with forecasting and the frequency with which impactful weather occurs require an active detect and response system. In this paper, we propose a dynamic, data-driven decision support system, with components of forecasting, realtime sensor observations, and route planning. We demonstrate our technology in the Dallas/Fort Worth metroplex, a large urban area with frequent thunderstorms which hosts the CASA Doppler radar network.
The high temporal and spatial resolution data provided by this network allows us to quickly and accurately identify ongoing meteorological hazards for flight planning purposes. Rapidly updating short term (0-90 minute) forecast data are generated with features extracted as obstacles to avoid. A flight path generator submits requests for path routing which include randomized start and end locations and times, weather tolerance parameters, and buffer zones. A customized obstacle course is created and used as the basis for routing. Weather processing workflows are instantiated with Mobius, a multicloud provisioning system. The Pegasus Workflow Management System orchestrates processing via scalable workload distribution to compute resources. Sensor data is transmitted and processed in real time, and routes are periodically calculated for proposed flights. A Google Maps front-end interface displays the weather features and flight paths. Herein, we focus on the overall system design, with particular emphasis on the dynamic flexibility and interoperability that our architecture allows.

Keywords: Cloud Computing · UAS · UAM · Doppler Radar · Flight Planning · DDDAS

⋆ Supported by the National Science Foundation (Awards #:1826997, 2018074, and 1700967) and the North Central Texas Council of Governments.

1 Introduction

The Urban Air Mobility (UAM) concept is rapidly advancing and could be a trillion dollar industry by 2040 [1]. Small aircraft are sensitive to weather conditions however, and the detection and avoidance of threatening meteorological features is a key safety consideration [2]. Weight restrictions may preclude on-board remote sensing equipment needed for proactive response. If industry operations are to expand beyond benign weather days, risk information will have to be provided before and during flights. As tolerances vary not just across aircraft types, but also rapidly changing aircraft states, the underlying Dynamic Data Driven Applications Systems (DDDAS) must be customized and adjustable on the fly. In this paper, we propose and implement such a system using live Doppler radar observations, deriving meteorological products and forecasts, and extracting areas of relevance on a per flight basis, based on declared risk parameters. We use a flight path simulator to inject proposed routes and a diverse and randomized set of weather requirements. We have developed a flight path routing algorithm to navigate obstacle courses, consisting of dynamic weather risk areas and static areas such as no fly zones. We also present an implementation of this system, which makes use of compute clouds and private, high-speed networks.

2 Background

Our work makes use of technologies and concepts that the team has developed over the course of many years.

CASA Doppler radar is a key technology for the detection and quantification of precipitation. Since 2013, the graduated NSF Engineering Research Center for Collaborative and Adaptive Sensing of the Atmosphere (CASA), has installed and operated a network of seven high resolution, X-band Doppler weather radars in the Dallas/Fort Worth (DFW) metroplex in North Texas [3]. These rapidly updating radars focus scans on the lowest portion of the atmosphere, providing observations near the ground where they are most relevant to people and low flying aircraft. Our DDDAS system leverages live CASA radar data for situational awareness.

DyNamo is an NSF Campus Cyberinfrastructure (CC*) integration project for creating a dynamic, network-centric platform for data-driven science. By coupling existing toolsets, the workflow management software suite Pegasus [4] and the high throughput computing framework HTCondor [5], and by developing the multi-cloud provisioning and monitoring tool Mobius [6], DyNamo has enabled new weather risk extraction workflows for CASA that were not previously possible [6]. Mobius allows provisioning virtual machine (VM) pools from multiple national-scale infrastructures like ExoGENI [7], and Chameleon [8], connecting them to the CASA data repository with private layer2 networks, and modulating provisioned bandwidths using virtual Software Defined Exchanges (SDX) [9].

DyNamo allows us to efficiently scale our flight path routing system as the number of flights increases and the weather deteriorates.

3 Dynamic Data Driven System for Urban Air Mobility and Weather

In our system (Fig. 1), routes are determined based on atmospheric observations, which are represented as dynamic weather risk obstacles. Such obstacles are created by workflows that operate on sensor data (radars and other weather sensors) and stored in the Obstacle database (Sect 3.2). The Flight database (Sect. 3.1) contains information about planned or current flights. Pathing requests are submitted into DyNamo's workflow management system for resource acquisition and load balancing.

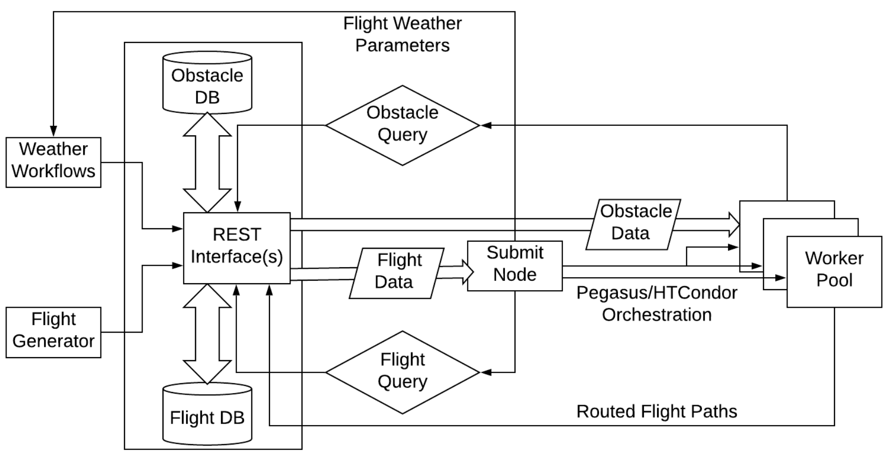

Fig. 1. Overall system architecture.

3.1 Flight Handling

Flight Database Interface. We implemented a Java based controller with a secure http based REST interface to a MySQL database. REST architectures with well defined API endpoints for information exchange are ubiquitous in the industry and reduce barriers to adoption. Flight descriptions are encoded as geoJSON [10], which combines the flexible and easily parsed JSON format, with well defined spatial GIS descriptors for location information. GeoJSON is natively supported by Google Maps which is our basis for flight and weather visualization. Flight information includes names, start and end points, waypoints (optionally), start time, weather parameter sets, and aircraft location if the flight is in progress. Our path routing function appends optimized path recommendations and the weather obstacle course on which the optimized path was generated back to the flight properties.

Flight Request Simulator. To evaluate our system, we have implemented a flight request simulator that mimics predefined flight requests. For demonstration purposes, we picked two random hospital locations in the DFW metroplex as start and end points, along with a randomized subset of discretized weather parameters for avoidance purposes.

3.2 Obstacle Handling

Obstacle Database Interface. Similar to the flight database interface, a Java based secure https REST interface has been created to receive and query geoJSON features representing areas of weather risk with certain representative characteristics. These can include magnitude or intensity thresholds, valid times ranging from the recent past into the future, or simple binary indicators such as the presence of hail. Static areas such as restricted airspaces around airports are also contained in the obstacle database. Areas of risk can be defined as closed concave polygons extracted from gridded datasets with contouring algorithms or simple point measurements.

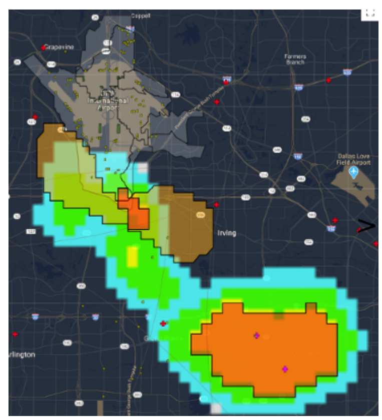

Fig. 2. Contours from CASA Wind (red) and CASA Hail (orange) products. Underlying raster image of CASA winds shown for reference. Static DFW airport contour also on display (clear) representing restricted air space.

Meteorological Products. Numerous meteorological feature detection algorithms contribute to populating the obstacle database (Table 1) and represent the primary variability of our DDDAS. Many are derived from the CASA radar data, but not exclusively. Each feature type has its own associated DyNamo workflow for product creation and/or extraction.[4]

Contouring. Feature extraction from gridded meteorological products relies heavily on contouring. Contour levels are determined dynamically by the superset of weather parameters from all flights in the database. For obstacle avoidance, we seek closed, ordered, concave polygons that can be treated as discrete objects. We have developed a C++ contouring class, implementing the Marching Squares

[4] Further details on workflows for weather product generation can be found in [6].

Product	Description
Rainfall Rates	Dual polarized radar data is converted to gridded quantitative precipitation estimates to determine rainfall rates [11].
Observed Winds	Radar velocity data is blended together to produce gridded estimates of the maximum observed wind speeds [6].
Reflectivity Nowcast	Multi-Radar reflectivity data is merged into an advection model producing Nowcast data, a predictive grid of Reflectivity valid 0-30 minutes into the future [12].
Reflectivity Forecast	Radar and other sensor data is fed into a full atmospheric data assimilation model called ARPS [13], producing a forecast from 30-90 minutes into the future, including resolving convective initiation.
Hail Detection	Multi-radar data are merged together and binary areas indicating the presence of hail are derived [14].
Lightning Detection	The Earth Networks lightning detection network [15] reports the location of cloud to ground lightning strikes.
NDFD Winds	The National Digital Forecast Database [16] produces a forecast of gridded 10m winds.
METARs	METeorological Aerodrome Reports are point based observations from weather stations located at airports across the region [17]. They report temperature, wind, atmospheric pressure, and cloud cover information.

Table 1. Weather products used to populate Obstacle Database.

algorithm [18] for generating isolines, and a stitching function to connect these together [19]. The results are encoded as geoJSON polygon features and posted into the obstacle database. Figure 2 depicts overlapping weather contours from the wind and hail workflows.

3.3 Flight Path Planning

Custom Querying. Once a minute our main control process queries the contents of the flight database, and creates a Pegasus job file for each active flight for route planning. These jobs are submitted to the HTCondor master node, which distributes them to the pool of available cloud worker nodes. Workers extract the time and weather parameters from the geoJSON properties of the flight and generate a set of queries to the obstacle database for the weather feature data of relevance, gradually constructing a full obstacle course. Queries include weather feature type, threshold magnitude, and radius for avoidance. Finally, static areas to avoid are added to the obstacle course.

Obstacle Buffering and Merging. Due to the fast developing nature of convective weather, forecast uncertainty, imperfect sensors and detection algorithms, processing latencies, and changing physical characteristics of different aircraft, we apply a convolution filtering algorithm for obstacle buffering on a per hazard basis, based on a flight's declared radius of avoidance for that hazard type.

Additionally, given the spatial correlations of various weather hazards, a substantial amount of overlap often exists among the weather feature obstacles. We therefore apply a concave polygon merging technique. A fine mesh grid is drawn across a bounding box containing the entire obstacle set. Then for each grid cell, we check whether it is inside or outside any of the obstacles, creating binary grid of inside/outside, and reapply the contouring algorithm described above to create a merged obstacle course.

Graphing and Path Routing Whereas ground traffic routing algorithms typically make use of roadway intersections as graph vertices, free air space provides infinite potential vertices, bound only by minimum incrementation. Therefore to simplify a search we make two assumptions:

(i) the optimal path from start to end is a straight line if no obstacles exist along the straight line path, and

(ii) if the straight line path intercepts an obstacle, the optimal path will include one or more convex hull points of the associated concave polygon obstacle.

With these in mind, the algorithm begins stepping small, fixed distances from the start point toward the end point, with each step checking for line intersection with an obstacle. Should one be encountered, we calculate the convex hull of that concave polygon using the Graham algorithm [20]. Then, from each of the convex points we recurse the algorithm to splay to the start and end points, repeating the splaying process with every new obstacle encountered. Splay paths can traverse along a vertex of the obstacle from which the convex hull point is associated, but cannot cross a vertex thereof, else it is eliminated from the graph. Ultimately, our graph is assembled consisting of all the splayed paths and the vertices of the concave buffered polygons themselves. Once the graph is created, we have implemented the A* ("A star") algorithm to evaluate it and return the shortest path among those defined in the graph [21]. Thereafter, a second optimization function is applied, necessary to shorten any concave traversals. For every waypoint in the route, we evaluate if a straight line path can be traveled to waypoints later in the route without crossing a polyline segment from polygons in our obstacle course, starting from the end working backward. If no obstacles are encountered, we remove all route segments in between. If a valid flight path is found, the newly proposed route is written back into the flight database along with the obstacle course used to create it. Figure 3 depicts unique obstacle sets and two routed paths.

3.4 Multi-Cloud Resource Management

Our on-demand data-driven system is realized as a series of scientific workflows associated with weather obstacle generation and flight path routing. A dedicated server located at the University of North Texas in Denton, TX serves as a weather data portal/repository and control interface. It uses Mobius [6] to provision the most resource intensive workflows on Chameleon when extra processing power is needed, whereas ExoGENI nodes are used for the bulk of the processing and can

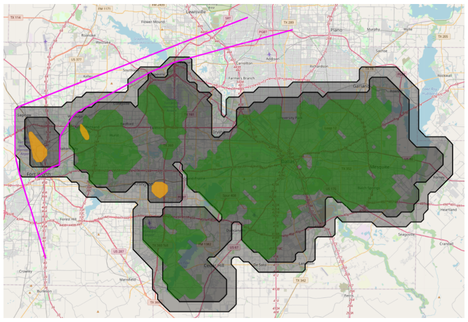

Fig. 3. Two path routed flights are shown as pink lines. Weather obstacles are depicted as green (10 minute CASA reflectivity nowcast) and orange (CASA observed hail) polygons. Two sets of black polygons represent customized buffer zones, based on per flight parameters. Flights also avoid DFW airport boundary (not shown).

be rapidly instantiated. Mobius allows us to connect to ExoGENI and Chameleon with layer2 networks and modulate bandwidth to individual workflows with SDX, prioritizing those associated with weather parameters of ongoing flights and thus extending the DDDAS concept to the networking layer.

Fig. 4 depicts data flows for weather processing workflows created by Mobius.

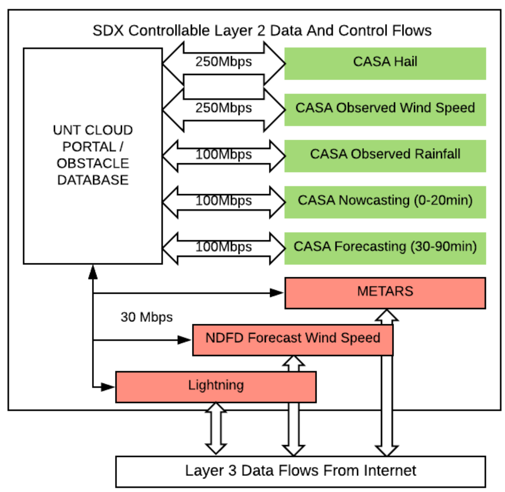

Fig. 4. Weather workflows are provisioned with layer2 networks and bandwidth modulated by virtual SDX. CASA based workflows shown in green, other sensors shown in red.

4 Conclusions

In this paper, we have described the framework of a complete, functioning DDDAS, operating with live sensor and model data that can be instantiated on demand. Weather geofence extraction occurs as a function of the declared tolerances of individul flight requests. The underlying compute and networking adapts to weather related load and flight monitoring needs in an automated fashion by modulating available bandwidth to weather workflows with SDX and distributing processing to pools of worker nodes with Pegasus and HTCondor. Flight planning is therefore unique and customizable for a given aircraft or aircraft class, with feedback into the underlying data generation. We believe this to be an effective design that accounts for the necessary considerations associated with a complex, multi-faceted system in a resource constrained environment. In future work we intend to formally evaluate the performance of the various subsystems.

References

1. Are flying cars preparing to take off? https://www.morganstanley.com/ideas/autonomous-aircraft (2019)
2. Nasa urban air mobility. https://www.nasa.gov/sites/default/files/atoms/files/uam-market-study-executive-summary-v2.pdf
3. McLaughlin, D.J., Chandrasekar, V.: Short wavelength technology and the potential for distributed networks of small radar systems. In: 2009 IEEE Radar Conference. pp. 1–3 (2009)
4. Deelman, E., et al.: Pegasus, a workflow management system for science automation. Future Generation Computer Systems 46, 17 – 35 (2015)
5. Epema, D., et al.: A worldwide flock of condors: Load sharing among workstation clusters. Future Generation Computer Systems 12(1), 53 – 65 (1996)
6. Lyons, E., et al.: Toward a dynamic network-centric distributed cloud platform for scientific workflows: A case study for adaptive weather sensing. In: 2019 15th International Conference on eScience (eScience). pp. 67–76 (2019)
7. Baldin, I., et al.: Exogeni: A multi-domain infrastructure-as-a-service testbed. In: McGeer, R., et al. (eds.) The GENI Book, pp. 279–315 (2016)
8. Keahey, K., et al.: Chameleon: a scalable production testbed for computer science research. In: Contemporary High Performance Computing, pp. 123–148 (2019)
9. Mandal, A., et al.: Toward prioritization of data flows for scientific workflows using virtual software defined exchanges. In: 2017 IEEE 13th International Conference on e-Science (e-Science). pp. 566–575 (2017)
10. The geojson format. https://tools.ietf.org/html/rfc7946
11. Chen, H., Chandrasekar, V.: The quantitative precipitation estimation system for dallas fort worth (dfw) urban remote sensing network. Journal of Hydrology 531, 259 – 271 (2015)
12. Ruzanski, E., et al.: The CASA Nowcasting System. Journal of Atmospheric and Oceanic Technology 28(5), 640–655 (05 2011)
13. Xue, M., et al.: The Advanced Regional Prediction System (ARPS), storm-scale numerical weather prediction and data assimilation. Meteorology and Atmospheric Physics 82(1-4), 139–170 (Jan 2003)
14. Chandrasekar, V., et al.: Dfw urban radar network observations of floods, tornadoes and hail storms. In: 2018 IEEE Radar Conference (RadarConf18). pp. 0765–0770 (2018)
15. Bui, V., et al.: A performance study of earth networks total lighting network (entln) and worldwide lightning location network (wwlln). In: 2015 International Conference on Computational Science and Computational Intelligence (CSCI). pp. 386–391 (2015)
16. National weather service - wind speed forecasts. http://www.weather.gov/ndfd/
17. Federal aviation administration: METAR/TAF. https://www.aviationweather.gov/metar
18. Maple, C.: Geometric design and space planning using the marching squares and marching cube algorithms. In: 2003 International Conference on Geometric Modeling and Graphics, 2003. Proceedings. pp. 90–95 (2003)
19. D3-contour. https://github.com/d3/d3-contour
20. Graham, R.L.: An efficient algorithm for determining the convex hull of a finite planar set. Inf. Process. Lett. 1, 132–133 (1972)
21. Hart, P., et al.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics SSC-4(2), 100–107 (1968)