Renewable Power Forecast Generation with Dask and Visualization with Bokeh

Antonio Lorenzo Assistant Research Scientist UA Dept. of Hydrology & Atmospheric Sciences



Will Holmgren, Asst. Research Professor Leland Boeman, Software Engineer "Wind and solar make up almost 50% of world electricity in 2050 - '50 by 50' and help put the power sector on track for 2 degrees to at least 2030." Bloomberg NEF New Energy Outlook 2019 https://about.bnef.com/new-energy-outlook/



BRIEF

US renewable energy transition to move faster than anticipated by 2022: FERC report

Dive Brief:

- By June 2022, the pace of U.S. renewables growth is going to surpass fossil fuel growth by a significantly greater margin than what FERC had anticipated as recently as April, according to the commission's <u>May 2019 Energy Infrastructure Update</u>, released Friday.
- The renewable energy-focused <u>SUN DAY Campaign said new</u> <u>renewable energy capacity would grow</u> more than 10% by 2022 while fossil fuel capacity would only increase about 1%, compared to the <u>April forecast of a 5% net increase</u>. The fossil fuel dip will be largely driven by the more than 4.6 GW of coal forecast for retirement, according to FERC's May update.
- While SUN DAY's analysis asserts that FERC "drastically revised" its 3-year forecast, "the generation additions/retirements section of the monthly report is NOT a forecast or prediction of Commission expectations," FERC media relations director Mary

DEEP DIVE

Solar + wind + storage developers 'gearing up' as hybrid projects edge to market

A "wave" of new projects is coming to use wind, solar, and battery storage in ways that will stabilize grids, increase efficiencies and lower power costs.

🗩 UTILITY DIVE Deep Dive Opinion Podcasts Library Events Jobs Topics ~

BRIEF

PNM proposes gas, solar+storage mix as cheapest option to replace San Juan coal plant



Power from solar plants can be highly variable due to clouds







A 20 MW ramp is about equivalent to the



Variable Power Output of a 28 MW PV Power Plant Power Output of a Gas Turbine 30 120 25 100 20 Power (MW) 80 Power (MW) 15 60 10 40 5 20 0 0 24:00 16:00 18:00 06:00 08:00 70:00 24:00 16:00 06:00 08:00 70:00 12:00 18:00 12:00 Local Time Local Time

Output from a gas turbine can be ramped quickly. Helps backup solar



Variable Power Output of a 28 MW PV Power Plant

Utilities are accustomed to controlling and scheduling their generators from a control room



Clouds/weather control the output of solar power plants



Utilities are accustomed to controlling and scheduling their generators from a control room



Forecasts allow Utilities to schedule other resources (batteries, market purchases, gas turbines, etc.) during expected drops in solar power.



UA Operational Forecasting Partners



Operational Forecasting for Utilities



https://forecasting.energy.arizona.edu/forecast/public









Weather Forecasting with WRF

- WRF: Weather Research and Forecasting
- Open-source mesoscale numerical weather model
- Run by many groups: NOAA, universities, etc.
- UA HAS runs WRF ~ 10 times daily
- NOAA NWP model (GFS, NAM, RAP) output for initial and boundary conditions

- Outer domain: 5.4 km, western CONUS
- Inner domain: 1.8 km, Arizona and New Mexico
- High resolution captures more topography
- http://www.atmo.arizon
 a.edu/?section=weather
 &id=wrf

ADVI: Arizona gridded Data VIsualization

- Interactive Bokeh visualizations of forecasts from WRF
- Helpful for model configuration and development



forecasting.energy.arizona.edu/advi

ARTSy: Arizona Rainfall Tracking System

- Bokeh visualization of MRMS precipitation estimates
- Used to verify WRF precipitation forecasts



Data from http://mrms.ncep.noaa.gov/data. Map tiles from Stamen Design. Plot generated with Bokeh by A. Lorenzo, W. Holmgren, M. Leuthold, UA HAS







nabu: renewable forecast generation with Dask

- Processes weather forecasts from WRF into wind and solar
 power forecasts
- Strategically utilizes Dask.delayed to parallelize operations
- Used for both production forecasts and research into improved forecasting methods
- In production, the distributed scheduler is used to run on multiple worker nodes
- Generates new forecasts for each utility every 5 minutes

Forecasts for all Utility renewable power plants



Forecasts for all Utility renewable power plants



Forecast for single power plant





def compute forecast future(name, unified_index, lat, lon, elev, db, table, wrf_vars, wrf_to_use, peak_power, units, *, delayed_config=None, config, **kwargs): raw_wrf = delayed(wrf.get_sensor_wrf_forecasts)(lat, lon, wrf_vars, wrf_to_use, config=delayed_config) clearsky = delayed(clearsky_profile)(name, lat, lon, elev, raw_wrf, unified_index, db, config=delaved config) clearsky_fx = delayed(time.interpolate_to)(clearsky, unified_index, config=delayed_config) wrf_power = delayed(convert_weather_to_power_pvlib)(name, lat, lon, elev, raw_wrf, unified_index, config=delayed_config) persistence_fx = delayed(persistence.compute_clearsky_persistence)(name, db, table, clearsky, unified_index, 'solarpower', units, config=delayed_config) ensemble = delayed(base.ensemble_forecasts)(wrf_power, unified_index=unified_index, persistence_fx=persistence_fx, smooth_pts=config.solarpower.ensemble_smooth_pts, config=delayed_config) combined = delayed(base.combine_forecast_dataframes)(wrf_power, ensemble, clearsky_fx, persistence_fx) scaled = delayed(base.scale_forecasts)(name, db, peak_power, combined, unified_index, config=delayed_config) return scaled

- Goal is to keep processing as a simple set of functions
- Instead of using Dask.Array/DataFrame, we wrap functions in Dask.delayed for lazy, parallelized evaluation



```
def compute forecast future(
     name, unified_index, lat, lon, elev, db, table,
    wrf_vars, wrf_to_use, peak_power, units, *,
     delayed_config=None, config, **kwargs):
raw_wrf = delayed(wrf.get_sensor_wrf_forecasts)(
     lat, lon, wrf_vars, wrf_to_use,
     config=delayed_config)
clearsky = delayed(clearsky_profile)(
     name, lat, lon, elev, raw_wrf, unified_index, db,
     config=delaved config)
clearsky_fx = delayed(time.interpolate_to)(
     clearsky, unified_index, config=delayed_config)
wrf_power = delayed(convert_weather_to_power_pvlib)(
     name, lat, lon, elev, raw_wrf, unified_index,
     config=delayed_config)
 persistence_fx = delayed(
     persistence.compute_clearsky_persistence)(
         name, db, table, clearsky, unified_index,
         'solarpower', units, config=delayed_config)
ensemble = delayed(base.ensemble_forecasts)(
     wrf_power, unified_index=unified_index,
     persistence_fx=persistence_fx,
     smooth_pts=config.solarpower.ensemble_smooth_pts,
     config=delayed_config)
combined = delayed(base.combine_forecast_dataframes)(
     wrf_power, ensemble, clearsky_fx, persistence_fx)
 scaled = delayed(base.scale_forecasts)(
      name, db, peak_power, combined, unified_index,
     config=delayed_config)
 return scaled
```

- Goal is to keep processing as a simple set of functions
- Instead of using Dask.Array/DataFrame, we wrap functions in Dask.delayed for lazy, parallelized evaluation



def compute forecast future(name, unified_index, lat, lon, elev, db, table, wrf_vars, wrf_to_use, peak_power, units, *, delayed_config=None, config, **kwargs): raw_wrf = delayed(wrf.get_sensor_wrf_forecasts)(lat, lon, wrf_vars, wrf_to_use, config=delayed_config) clearsky = delayed(clearsky_profile)(name, lat, lon, elev, raw_wrf, unified_index, db, config=delaved config) clearsky_fx = delayed(time.interpolate_to)(clearsky, unified_index, config=delayed_config) wrf_power = delayed(convert_weather_to_power_pvlib)(name, lat, lon, elev, raw_wrf, unified_index, config=delayed_config) persistence_fx = delayed(persistence.compute_clearsky_persistence)(name, db, table, clearsky, unified_index, 'solarpower', units, config=delayed_config) ensemble = delayed(base.ensemble_forecasts)(wrf_power, unified_index=unified_index, persistence_fx=persistence_fx, smooth_pts=config.solarpower.ensemble_smooth_pts, config=delayed_config) combined = delayed(base.combine_forecast_dataframes)(wrf_power, ensemble, clearsky_fx, persistence_fx) scaled = delayed(base.scale_forecasts)(name, db, peak_power, combined, unified_index, config=delayed_config) return scaled

- Dask optimizes the task graph to avoid duplicate computations and to order the computations
- Saving the Dask graph along with python/Docker versions provide some provenance for the computation
- Utilize other Dask integrations (e.g. dask- jobqueue) to run many hindcasts in parallel for research

Performance evaluation with the Dask Profiler

- Tasks graphs are collected, optimized and sent to a distributed scheduler
- Workers run in a Kubernetes cluster
- Adjust function scope to avoid excessive IO as tasks compute



Forecast output visualization



https://forecasting.energy.arizona.edu/forecast/public





Solar Forecast Arbiter



- Open-source framework for solar forecast evaluations that are impartial, repeatable, and auditable.
- \$1M in funding from the Department of Energy Solar Forecasting 2 program
- Allows forecasters to compare their forecasts against a set of benchmarks, and allows utilities to compare many forecasts
- Ultimately lowers cost of solar power
- https://solarforecastarbiter.org
- https://github.com/solararbiter
- https://dashboard.solarforecastarbiter.org

Thank you!

















forecasting.energy.arizona.edu

github.com/uarenforecasting