# Design and Analysis of Algorithms (24CS2203)

# High-Speed Weather Forecasting via PRAM-Based Parallel Computation

Jagan Sunkavalli: 2420030714

Pardheev Nalla: 2420090145

**Course Instructor**

**Dr. J Sirisha Devi**

**Professor**

**Department of Computer Science and Engineering**

# Case study - statement

- Weather forecasting is a complex scientific task requiring the processing of massive meteorological data for timely and accurate predictions.

- Traditional serial computational methods often fall short due to the high computational complexity of numerical weather prediction models.

- This case study focuses on leveraging Parallel Random-Access Machine (PRAM) algorithms to parallelize meteorological

# Algorithm

- Divide meteorological data into blocks, assign each to a processor.
- Processors read data concurrently into shared memory.
- For each time step, processors compute updates in parallel using atmospheric equations.
- Integrate new observational data periodically with synchronization.
- Apply parallel spectral transformations (FFT) on assigned data.
- Optionally, run multiple ensemble forecasts simultaneously.
- Aggregate all processor results into the final weather forecast.

# Pseudo code

Input: MeteorologicalData[N][M], TimeSteps, NumProcessors P
Output: ForecastData[N][M][TimeSteps]

```
InitializeGrid(MeteorologicalData, P)
Parallel For each processor p in [1..P]:
    Assign sub-grid to processor p
    Load assigned data block into shared memory

For t = 1 to TimeSteps:
    Parallel For each processor p in [1..P]:
        Compute atmospheric state updates (u,v,T,ρ,q) using PDEs
        Write results concurrently to shared memory

    If new observation at time t:
        Parallel For each observation:
            Assimilate data with Kalman filter corrections
            Concurrently write corrections to shared memory

    Parallel For each latitude band:
        Perform FFT and Legendre Transform for spectral analysis
        Write coefficients to shared memory

    Parallel Ensemble Forecast (optional):
        Parallel For each ensemble member:
            Perturb initial conditions and run forecast
        Compute ensemble mean and spread
    Return ForecastData
```

# Time Complexity

•**Sequential Time**:

$$O(n \times T)$$

Each of the $n$ grid cells is updated for every time step $T$ sequentially.

•**Parallel PRAM Time**:

$$O\left(\frac{n \times T}{P}\right) + O(\log P)$$

Computation is divided among $P$ processors with synchronization overhead $O(\log P)$.

•**Interpretation**:

Increasing the number of processors $P$ reduces the computation time approximately by a factor of $P$, yielding significant speedup compared to the sequential approach.

# Space Complexity

- **Input Data Storage**: $O(n)$
  Stores meteorological variables for $n$ atmospheric grid cells.
- **Processor State Memory**: $O(P)$
  Local variables and temporary storage for each of the $P$ processors.
- **Shared Memory**: $O(n)$
  Shared access memory for concurrent processor operations on atmospheric data.
- **Overall Space Complexity**:

$$O(n) + O(P) \approx O(n + P)$$