Koneru Lakshmaiah Education Foundation
(Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
Off-Campus: Bachupally-Gandimaisamma Road, Bowrampet, Hyderabad, Telangana - 500 043.
Phone No: 7815926816, www.klh.edu.in

Department of Computer Science and Engineering

2024-2025
Even Semester

DESIGN AND ANALYSIS ALGORITHMS (24CS2203)

ALM – PROJECT BASED LEARNING

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

PROBLEM STATEMENT

The problem of weather forecasting involves predicting atmospheric conditions by processing large-scale meteorological data using computational models.

Given:

- Complex coupled **nonlinear partial differential equations** modelling atmospheric behaviour
- **High dimensional meteorological data** involving variables like temperature, pressure, humidity, wind speed, etc.
- Need for **accurate and timely forecasts** for practical decision-making

Goal:

- Develop **parallel algorithms using the Parallel Random-Access Machine (PRAM)** model to process weather data rapidly
- Decompose **meteorological computations** into **concurrent tasks** suitable for PRAM's shared memory parallelism
- **Reduce overall computation** time while maintaining or improving accuracy of weather forecasts

Challenges:

- Extremely high computational complexity of numerical weather prediction
- Large data volume requiring efficient memory and processing management
- Synchronization and communication overhead between parallel processors
- Handling of dynamic data assimilation and ensemble forecasting

Objectives:

- Understand PRAM architecture and its applicability to weather forecasting algorithms
- Design and implement parallel algorithms for atmospheric equation solving, spectral transformations, and data assimilation
- Analyse time and space complexity benefits of parallel PRAM computation versus serial processing
- Evaluate theoretical speedup and real-world feasibility of PRAM-based weather forecasting

Applications:

- Real-time regional and global weather prediction systems
- Environmental monitoring and disaster preparedness
- Ensemble forecasting used by meteorological centres such as ECMWF, NCEP

ALGORITHM / PSEUDO CODE

This section describes our parallel algorithm for rapid meteorological data processing in weather forecasting, based on the PRAM model.

The approach breaks down atmospheric computation tasks so they can be processed simultaneously.

Inputs

- **Meteorological Data Array:** atmospheric measurements (temperature, pressure, humidity, wind, etc.)
- **Parameters for simulation**: grid size, time steps, model constants
- **Number of Processors (P):** available for parallel computation

Steps

1. **Initialization:**
    - Partition meteorological data into P blocks.
    - Assign each block to a processor.

2. **Parallel Data Loading:**
    - All processors read assigned data into shared memory concurrently.

3. **Parallel Computation:**
    - For Time Step t = 1 to T:
        - Each processor computes updates for its data block based on forecast equations (e.g., Navier-Stokes, temperature, pressure).
        - Shared parameters and neighbouring block data are accessed using PRAM's shared memory model.

4. **Data Assimilation (if needed):**
    - At designated intervals, processors update blocks with new observations.
    - Synchronize shared memory for assimilated values.

5. **Spectral Transformations:**
    - Each processor applies relevant transforms (e.g., FFT) on its block for spectral analysis.

6. **Ensemble Forecasting (optional):**
    - Run multiple simulations in parallel, each representing different initial conditions.

7. **Aggregation:**
    - Collect results from all processors and aggregate into a unified forecast output.

Output

- Updated forecast data for future time steps, ready for visualization or further analysis.

Example

Suppose we want to predict temperature for 1,000 grid cells over 100-time steps:

- Input: array of temperature and atmospheric measurements for 1,000 cells
- Number of processors: P = 10
- Each processor receives 100 cells and processes updates for all time steps simultaneously
- After all steps, processors share and synchronize their results in shared memory for final aggregation

TIME COMPLEXITY

Our Weather Forecasting Algorithm

Aspect	Complexity	Explanation
Input size	n (Grid cells in atmospheric model)	Number of total data points being processed
Time steps	T	Number of forecast intervals processed
Number of processors	P	Processors working in parallel

Sequential Algorithm:

$$\boldsymbol{O(n \times T)}$$

Every grid cell is updated for all time steps one after another.

Parallel PRAM Algorithm:

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

Grid cells are divided among P processors. The overhead $O(\log P)$ is for synchronization and communication.

SPACE COMPLEXITY

Aspect	Explanation
Input Data Storage	$O(n)$
Processor State and Memory	$O(P)$ - Each processor maintains local state and temporary variables.
Shared Memory	$O(n)$ - For shared meteorological variables accessed by all processors.

Overall Space Complexity Formula:

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

- The atmospheric data dominates the space requirement.
- Space increases slightly with more processors due to local memory usage.
- PRAM's shared memory model facilitates concurrent access without duplicating data unnecessarily.

CONCLUSION

- Parallel computation using the PRAM model significantly accelerates meteorological data processing for weather forecasting.
- PRAM algorithms effectively decompose complex atmospheric calculations into concurrent tasks, enhancing computational efficiency.
- Time complexity improves approximately by a factor equal to the number of processors, achieving substantial speedup over sequential methods.
- Space complexity remains dominated by data size, with parallelization adding minimal overhead.
- This approach enables scalable, high-resolution weather prediction suitable for real-world forecasting centres.
- PRAM-based parallel algorithms provide a practical foundation for future advancements in numerical weather prediction and environmental modelling.

GitHub Repository Link:

https://github.com/Jagan-Dev-9/DAA_TEAM-10

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

<u>Input</u>: MeteorologicalData[N][M], TimeSteps, NumProcessors P
<u>Output</u>: 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**:

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