PARALLEL IMPLEMENTATION OF LOGISTIC REGRESSION USING GRADIENT DESCENT

CSE 633 PARALLEL ALGORITHMS

SWAPNIL SUSHIL PANDEY

University at Buffalo The State University of New York

What is Logistic Regression?

- Logistic regression is a classification model
- It is a process of modeling the probability of a discrete outcome given an input variable
- The outcome is often binary in nature, but this can also be used for multinomial classification
- It is useful to determine what category would a new sample best fit into

Some Real-Life Applications of Logistic Regression

Medical Research

- Medical researches need to find out how exercise could impact the probability of a heart attack
- This is an example of binomial classification wherein the possible outcomes are
  - The patient will likely get a heart attack
  - The patient will likely not get a heart attack

Credit or Debit Card Frauds

- For any given transaction that occurs, it is the responsibility of the provider to evaluate whether the transaction was fraudulent.
- The transaction amount, credit score, usage location and purchase history are some of the factors that can be used to determine the outcome
- This once again is an example of binomial classification with the possible outcome being:
  - Fraudulent transaction
  - Nonfraudulent transaction

Why do we need to parallelize the process?

The Data Usage Statistics of Some Mainstream Websites

- Facebook generates 4 petabytes of data per day
- Google search crawler handles 850 TB of data raw from the web
- Twitter generates more than 12 terabytes of data per day
- There are over 5 billion snaps(photos & videos) created on Snapchat everyday

The Data Usage Statistics of Some Mainstream Websites

- Facebook generates 4 petabytes of data per day
- Google search crawler handles 850 TB of data raw from the web
- Twitter generates more than 12 terabytes of data per day
- There are over 5 billion snaps(photos & videos) created on Snapchat everyday

Processing these large amounts of data

- All the websites are among the primary sources of information, communication or entertainment for a huge number of people
- It is important that the information being provided is accurate and the content provided to users is relevant to them
- All this involves processing of huge amounts of data
- Parallelizing the tasks massively boosts the processing speed

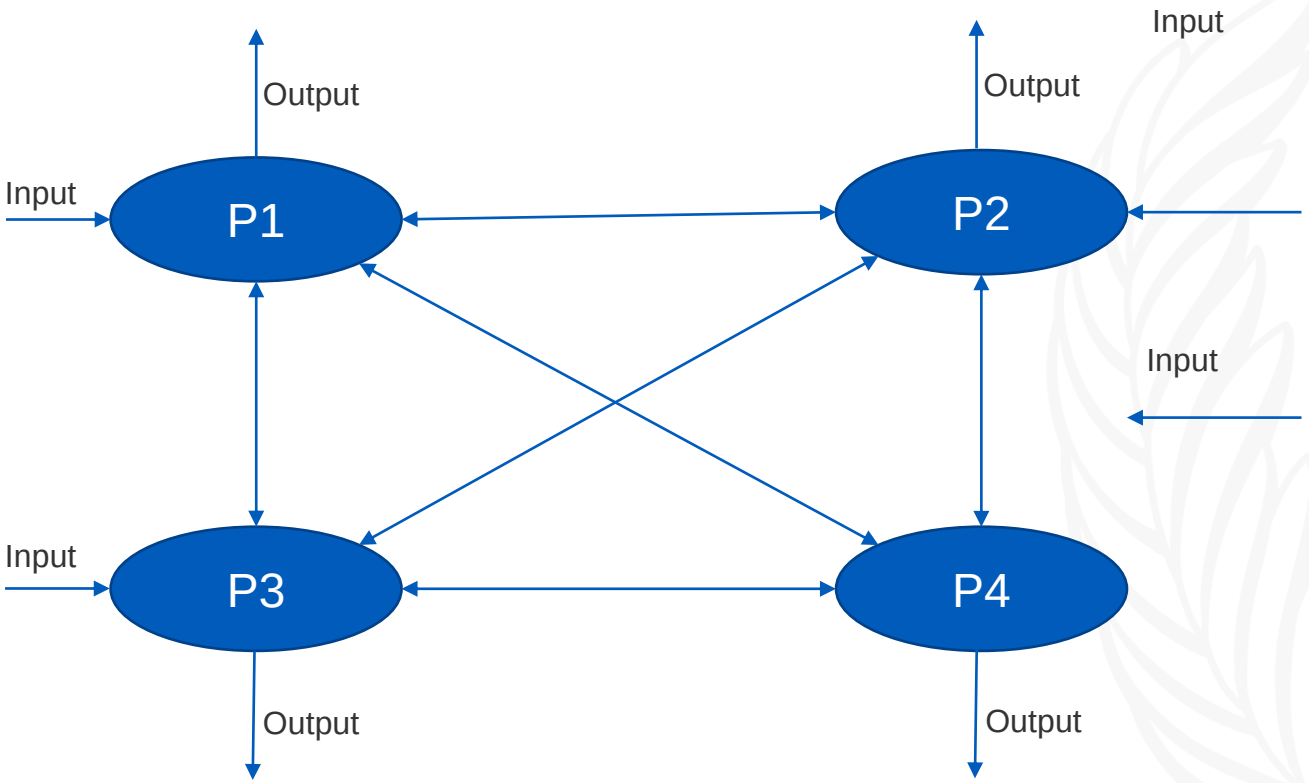
Parallelization Approach

- The gradient descent calculation process involves a lot of computation
- This is the focus of the parallelization process
- Data is uniformly distributed among all the processors
- Each processor performs gradient descend on its data set
- They broadcast the calculated gradient values to all other processors in the system
- Each processor uses the received values to update the local weights
- This process is repeated till the gradient converges or the number of epochs has been met

Gradient Calculation

- The convergence condition is set at 0.1% change and the number of epochs is set to 1000
- Numpy arrays of length 8,388,608 were provided as input
- The time complexity for the calculation is of the order O(n), where n is the size of the input

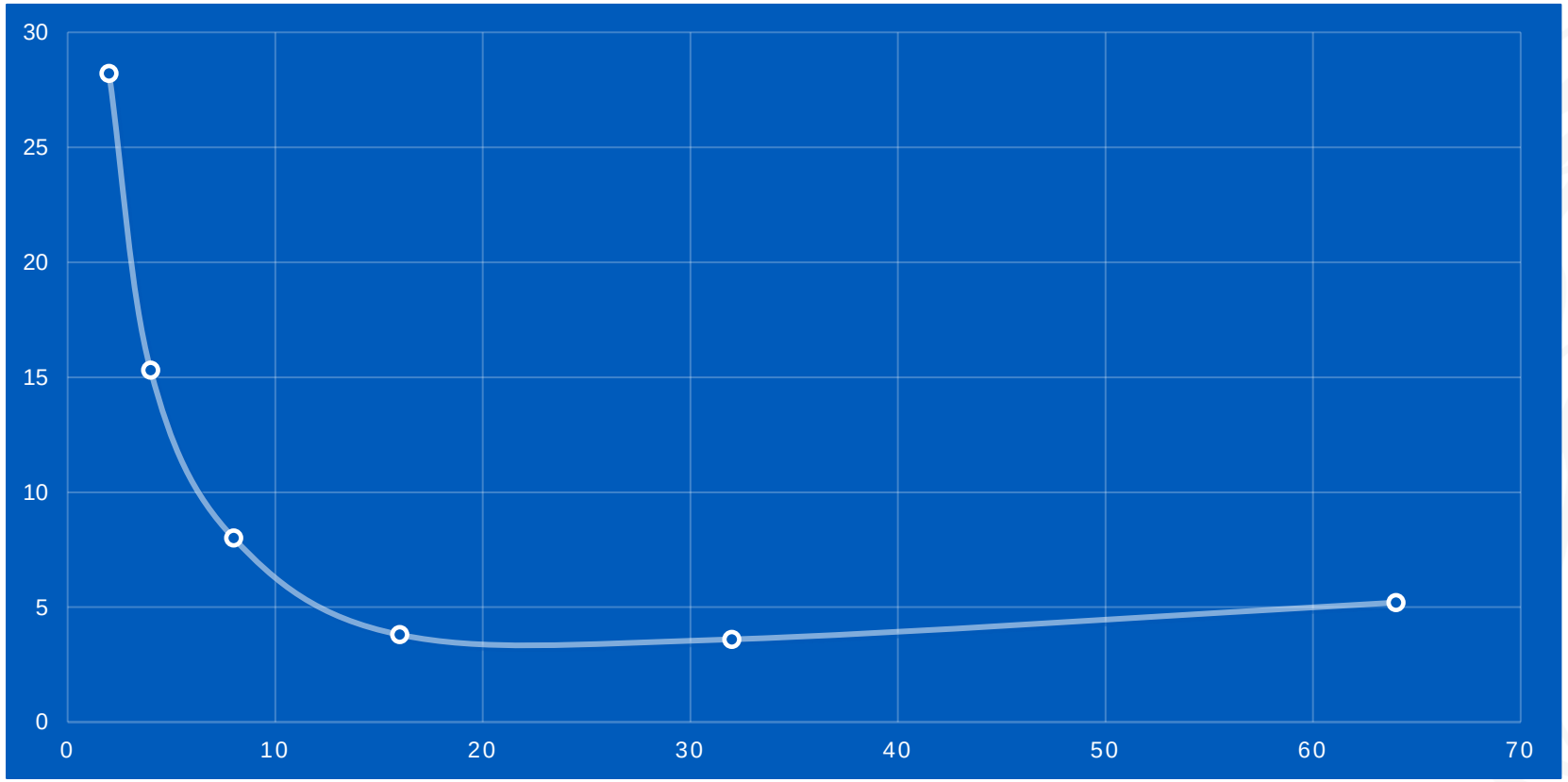
Communication between processors



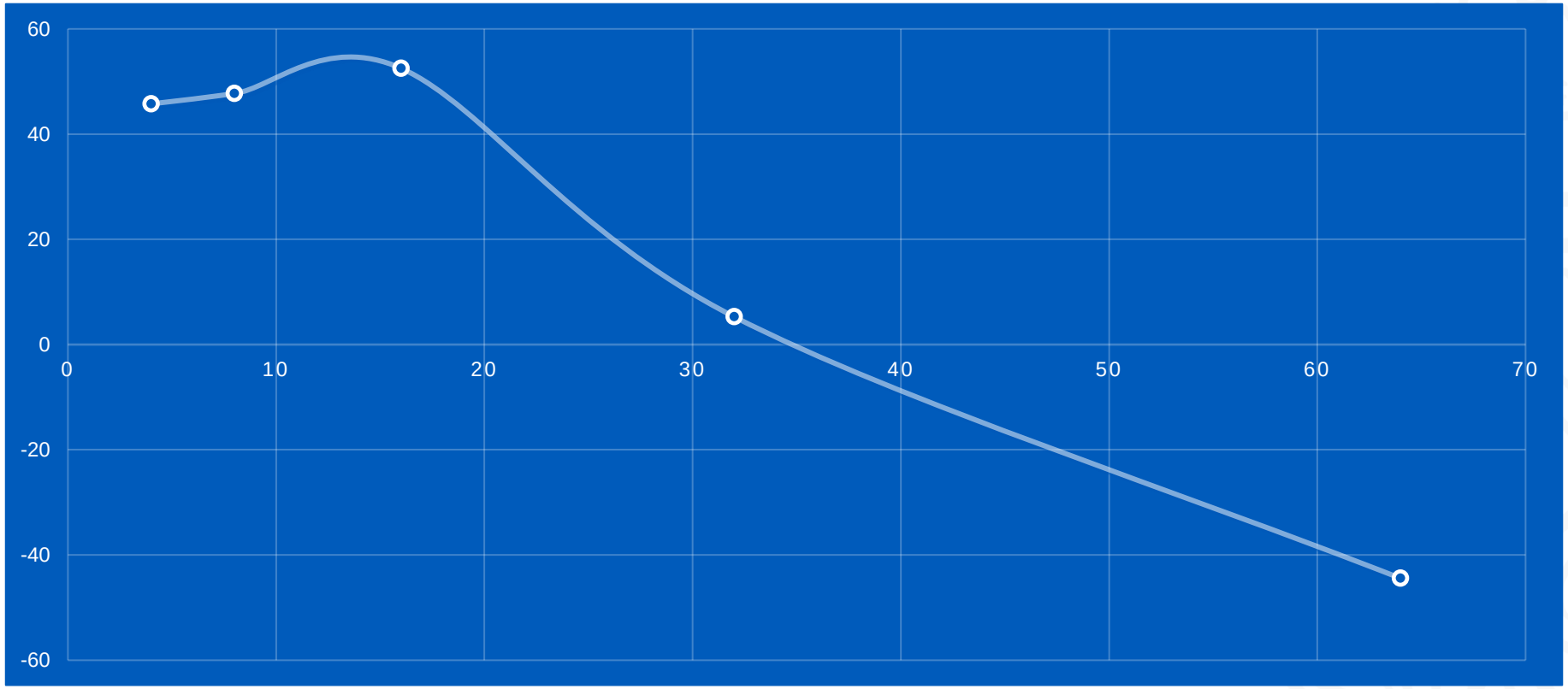
Results

Number of Processors	Total Running Time
2	28.2 s
4	15.3 s
8	8.0 s
16	3.8 s
32	3.6 s
64	5.2 s

Nodes vs Time Comparison



Nodes vs Relative Speedup Comparison



Conclusion

- The time required for gradient calculation decreased linearly with the increase in the number of processors
- This was in line with the O(n) time complexity of the calculation function
- After a certain number of nodes however, the communication overhead overshadowed the increase in efficiency obtained from parallelizing the task

THANK YOU

University at Buffalo The State University of New York