

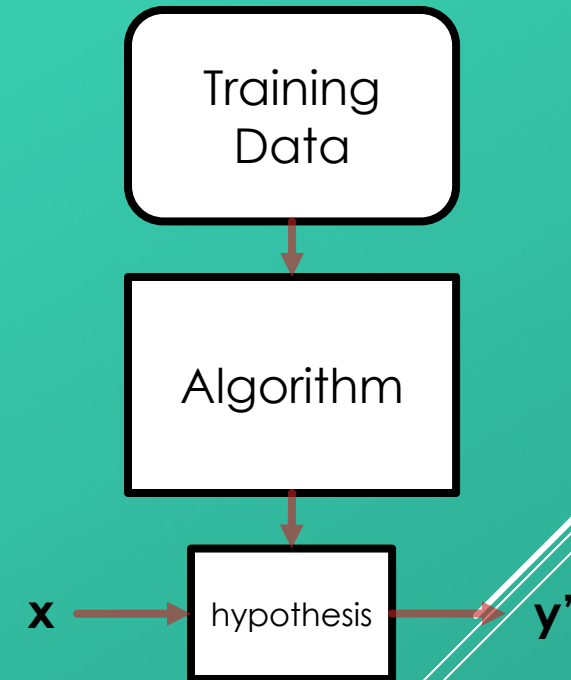
PARALLELIZED IMPLEMENTATION OF LOGISTIC REGRESSION USING MPI

CSE 633 PARALLEL ALGORITHMS

BY PAVAN G JOSHI

What is machine learning?

- ▶ Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed.^[1]
- ▶ Machine learning focuses on the development of computer programs that can change when exposed to new data.^[1]
- ▶ Easier to make machines learn real life examples rather than explicitly write real life rules

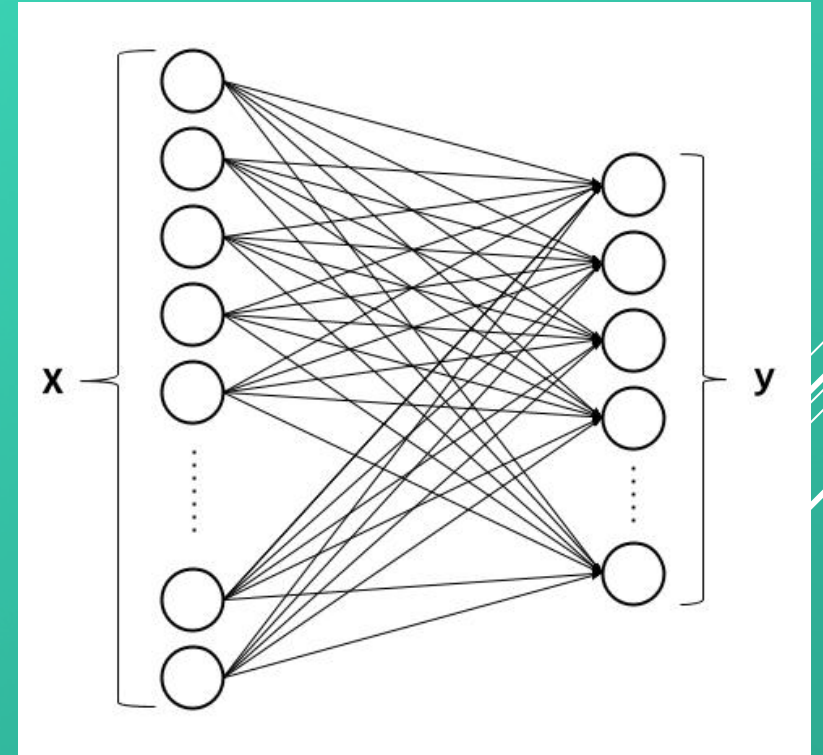


What is Logistic Regression?

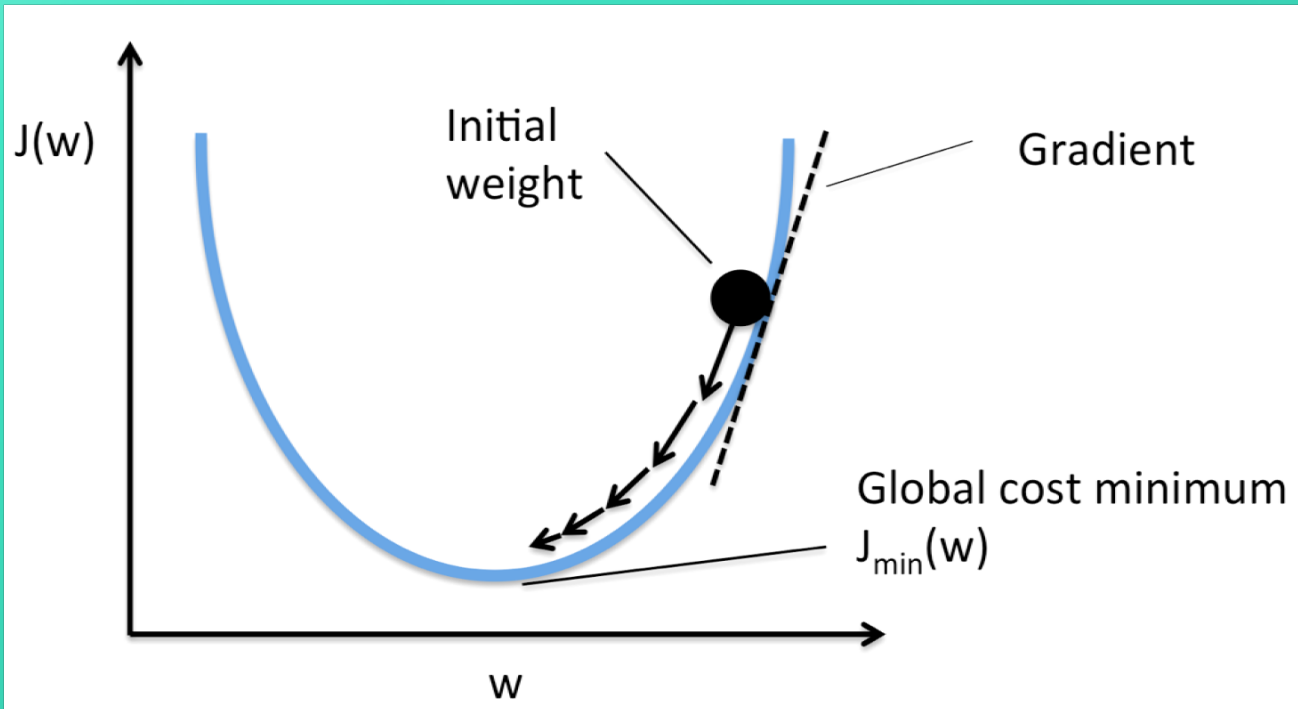
- ▶ Regression involves estimating the relationship between variables/features and dependent variable.
- ▶ Logistic Regression is a form of supervised learning algorithm where the ground truths are fed to the algorithm along with the features. The algorithm learns the relationship between the features and the ground truths and can help predict the classes/categories of unseen data/features.
- ▶ Requires the use of optimization algorithms such as gradient descent to get the best estimation of the relationship.
- ▶ Parallelization of Logistic Regression requires parallelization of optimization algorithms

Logistic Regression

- ▶ We are basically trying to fit an equation $y = g(x\theta)$, here $g(.)$ is the activation function.
- ▶ Logistic Regression involves:
 - ▶ Initializing weights θ randomly. Also we need to initialize a learning parameter α and a regularization parameter λ .
 - ▶ Gradient Descent - Compute the gradients and update the weights according to the learning parameters. Repeat the steps till convergence or till a preset number of epochs or iterations
 - ▶ Perform validation and predict the values.



Gradient descent



- ▶ Gradient descent is a first-order iterative optimization algorithm.
- ▶ To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point.

Gradient Descent – Algorithm

Repeat until convergence

{

$$\theta_{jk} := \theta_{jk} - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

For each j

}

- ▶ All θ should be updated simultaneously
- ▶ Total number of computations required in each iteration depends on
 - ▶ $m \rightarrow$ Number of samples in the training dataset
 - ▶ $j \rightarrow$ Number of features in each sample
 - ▶ $k \rightarrow$ Number of categories/classes

Why Parallelize?

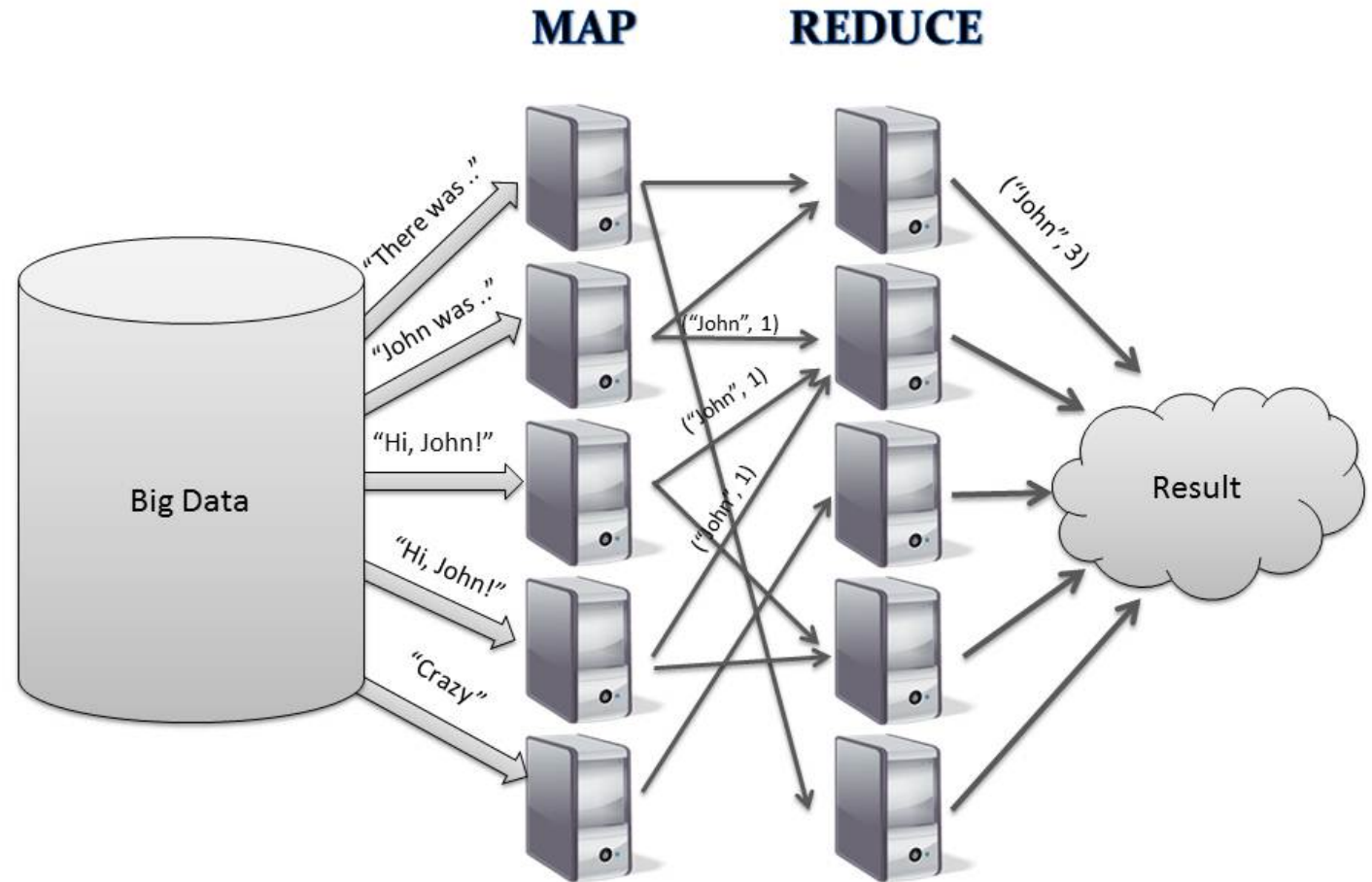
- ▶ Data Explosion
 - ▶ Google grew from processing 100TB a day in 2004 to 20PB a day in 2008.
 - ▶ Facebook claims to store upwards of 300PB with an increase of about 600TB daily.
- ▶ Logistic Regression involves optimization which can involve large computations.

What is MapReduce?

- ▶ MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.
- ▶ It was created by Google, Inc. in 2004 to process large scale data that was obtained from the world wide web.
- ▶ The core idea behind MapReduce is mapping your data set into a collection of <key, value> pairs, and then reducing over all pairs with the same key.

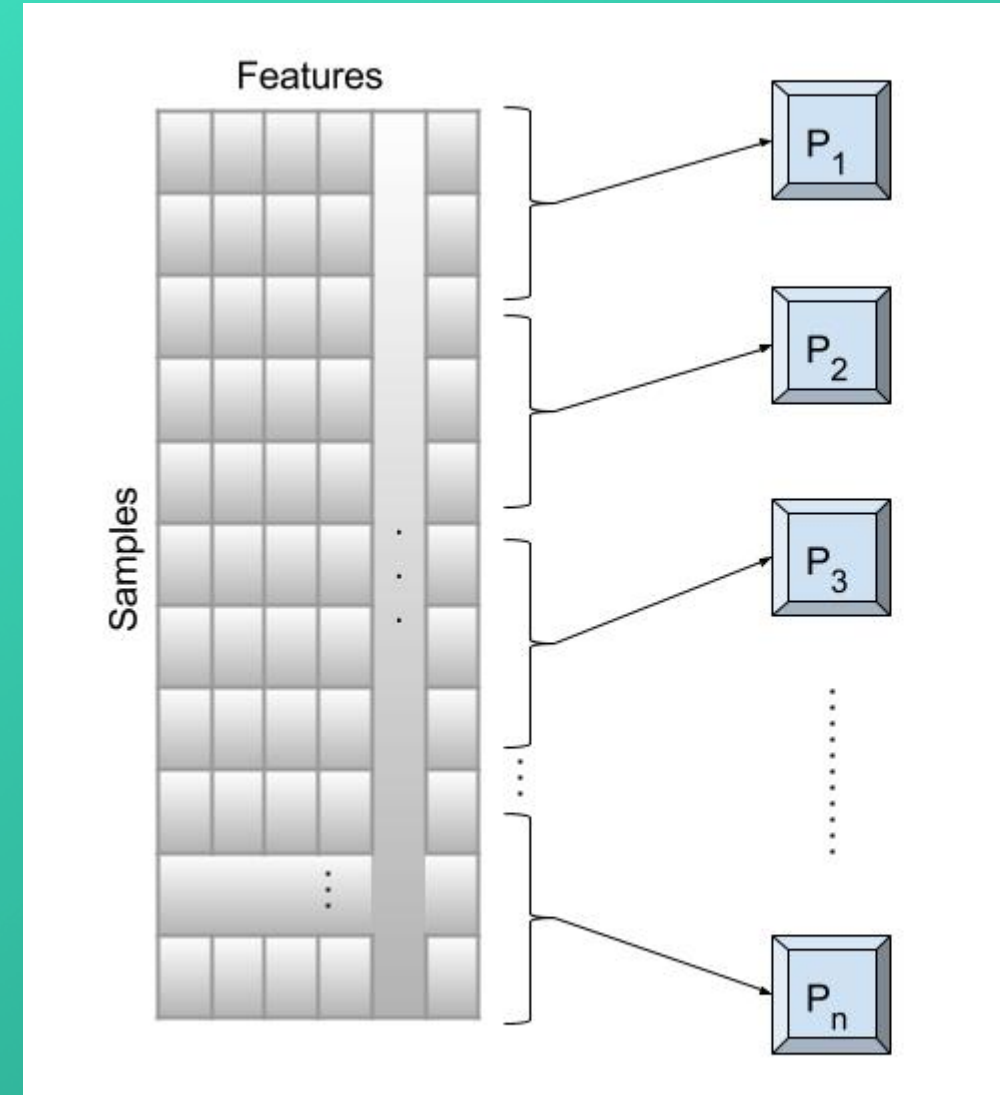
MapReduce

- ▶ A MapReduce program is composed of a `Map()` procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a `Reduce()` method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies).



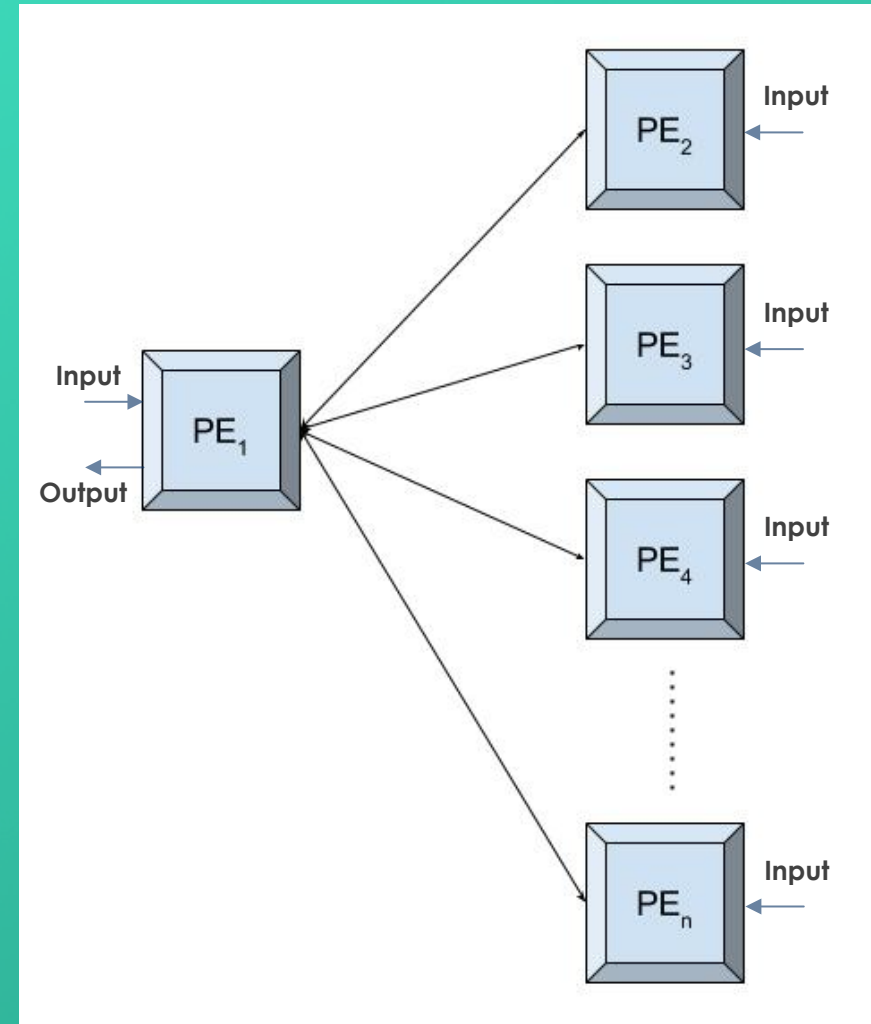
Implementation Approach

- ▶ Parallelization is implemented by dividing the data between the processors
- ▶ Each processor is responsible for a particular subset of samples.
- ▶ The respective subset of data is distributed to each processor.
- ▶ Each processor performs gradient descent on its set of samples locally



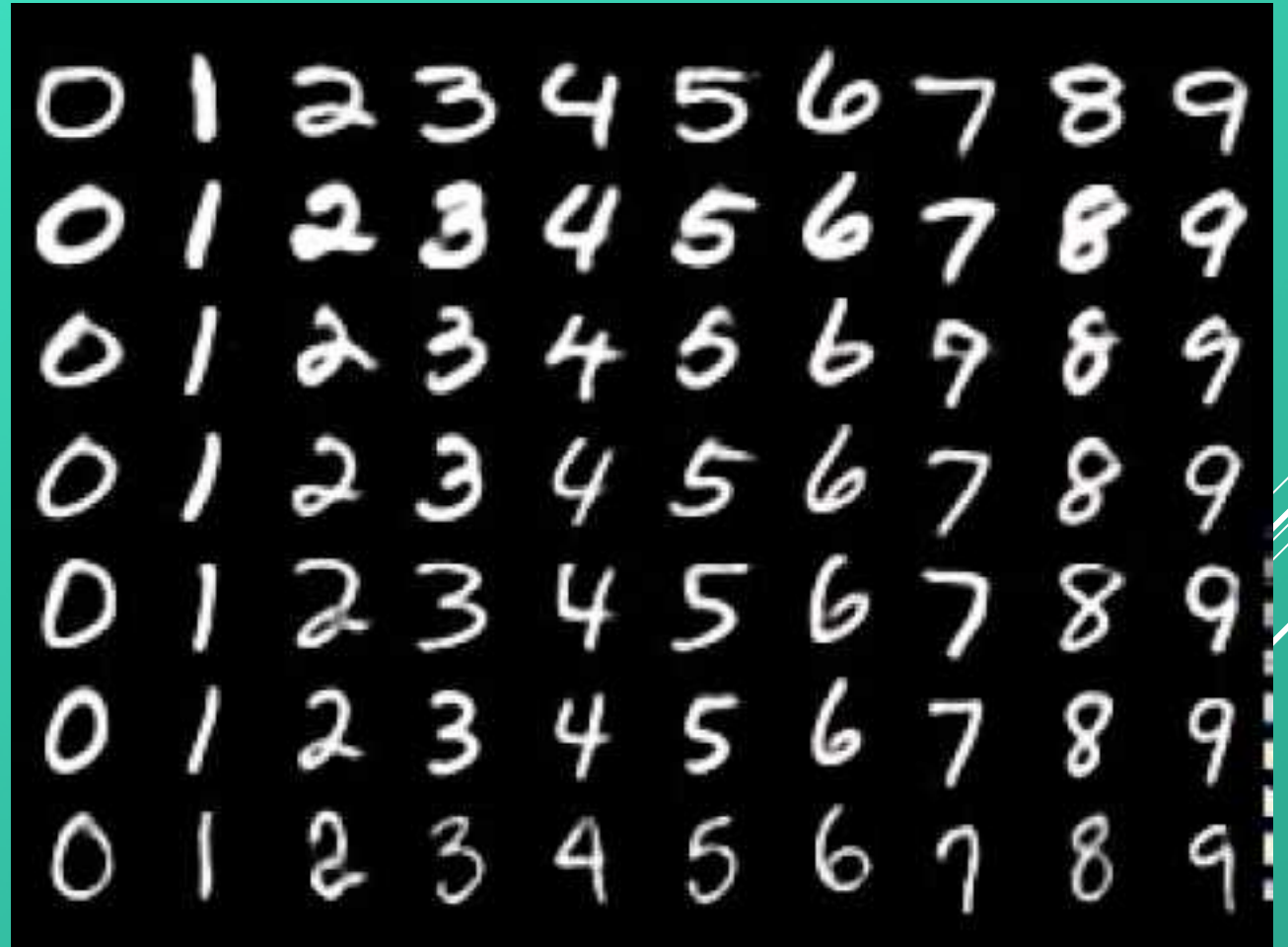
Implementation Approach

- ▶ Each processor computes the gradients locally on its set of samples.
- ▶ The gradients computed is then propagated to a master node (PE_1) which aggregates the gradients and updates the weights.
- ▶ The master node (PE_1) then broadcasts the updated weights to all the PE's in the system.
- ▶ The entire process is repeated until gradient convergence or till the number of epochs set has been met.

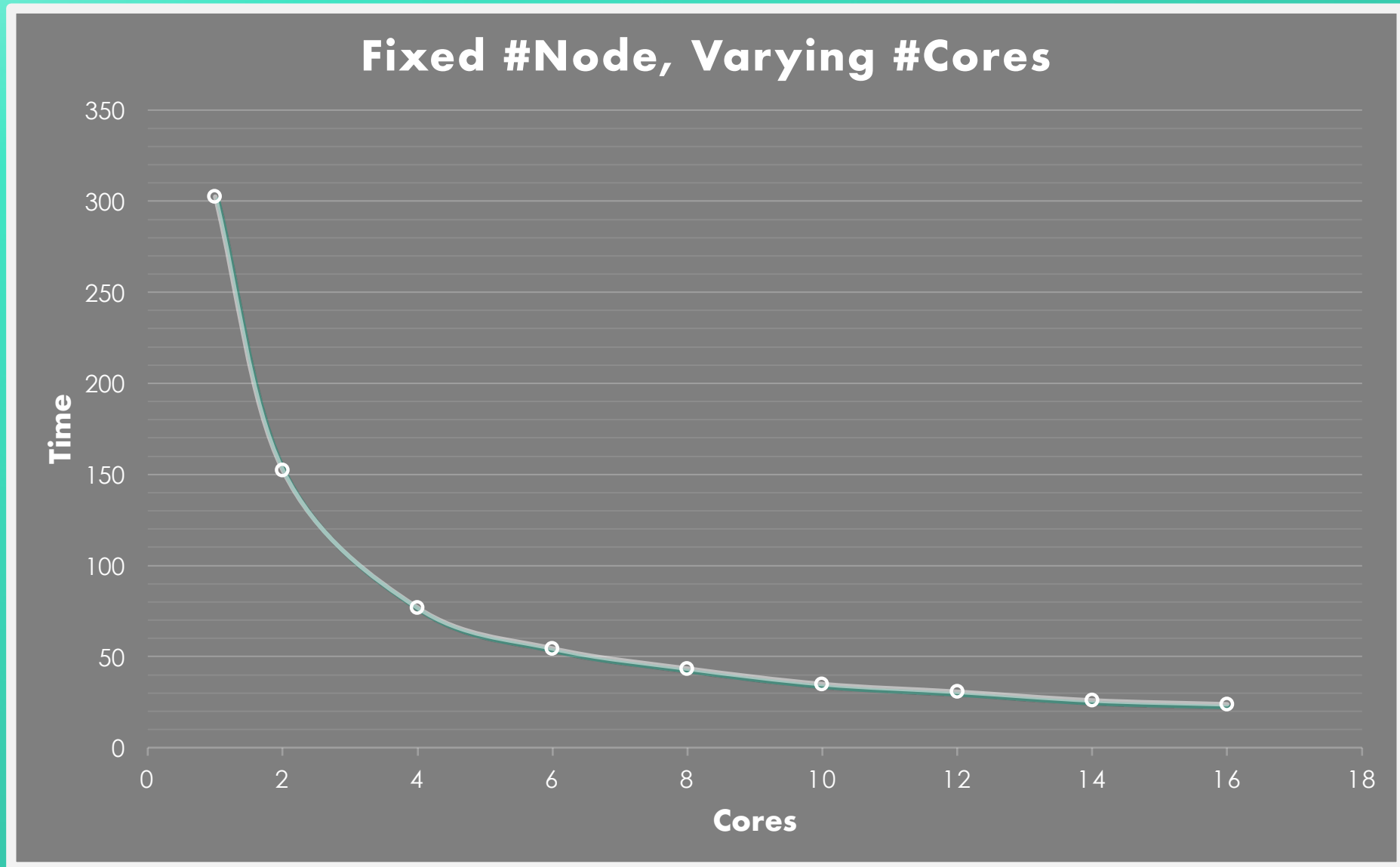


Dataset

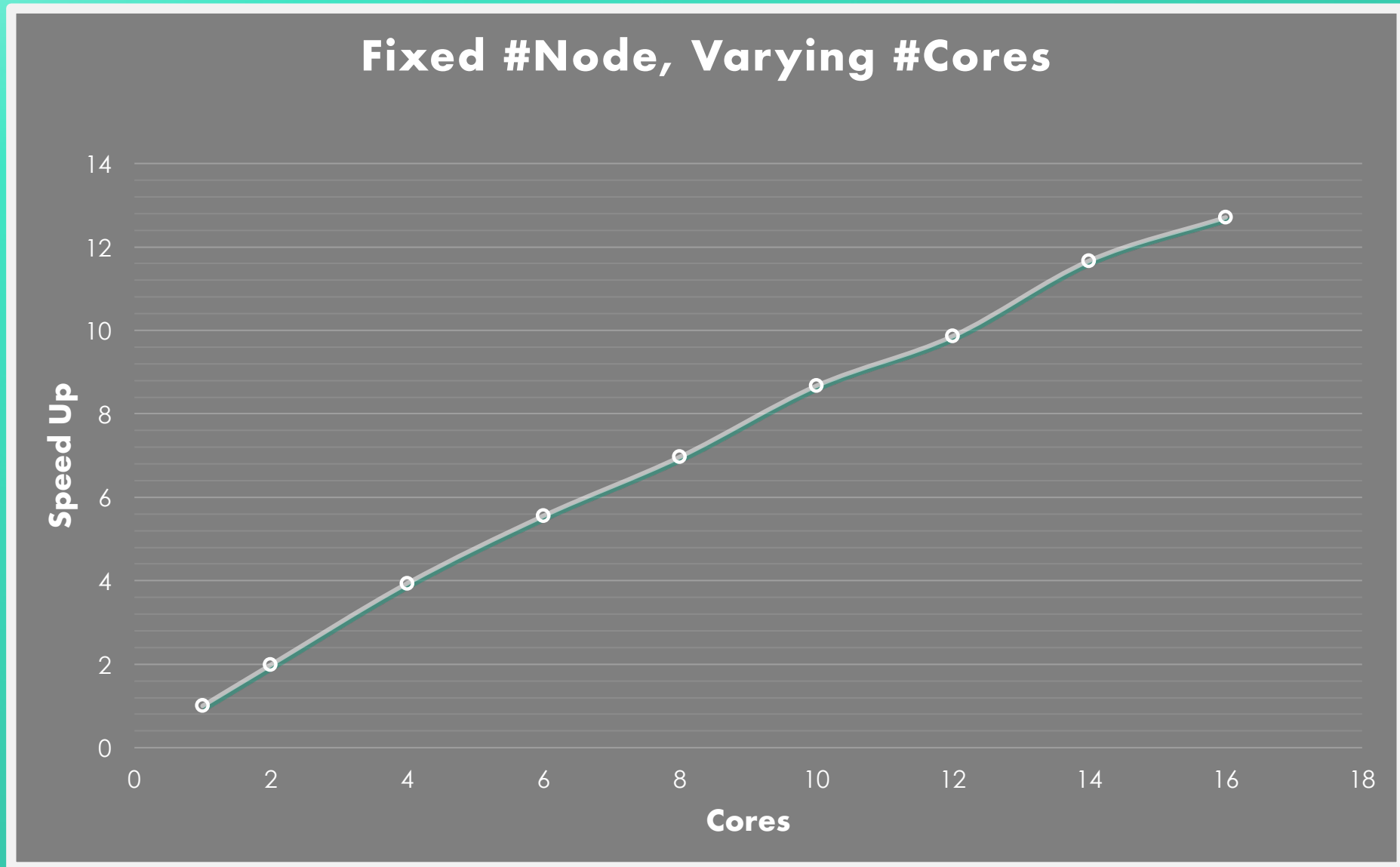
- ▶ MNIST Handwritten Dataset
- ▶ Contains a total of 70000 samples of images along with labels
- ▶ Each image has a resolution of $28 \times 28 = 784$ pixels



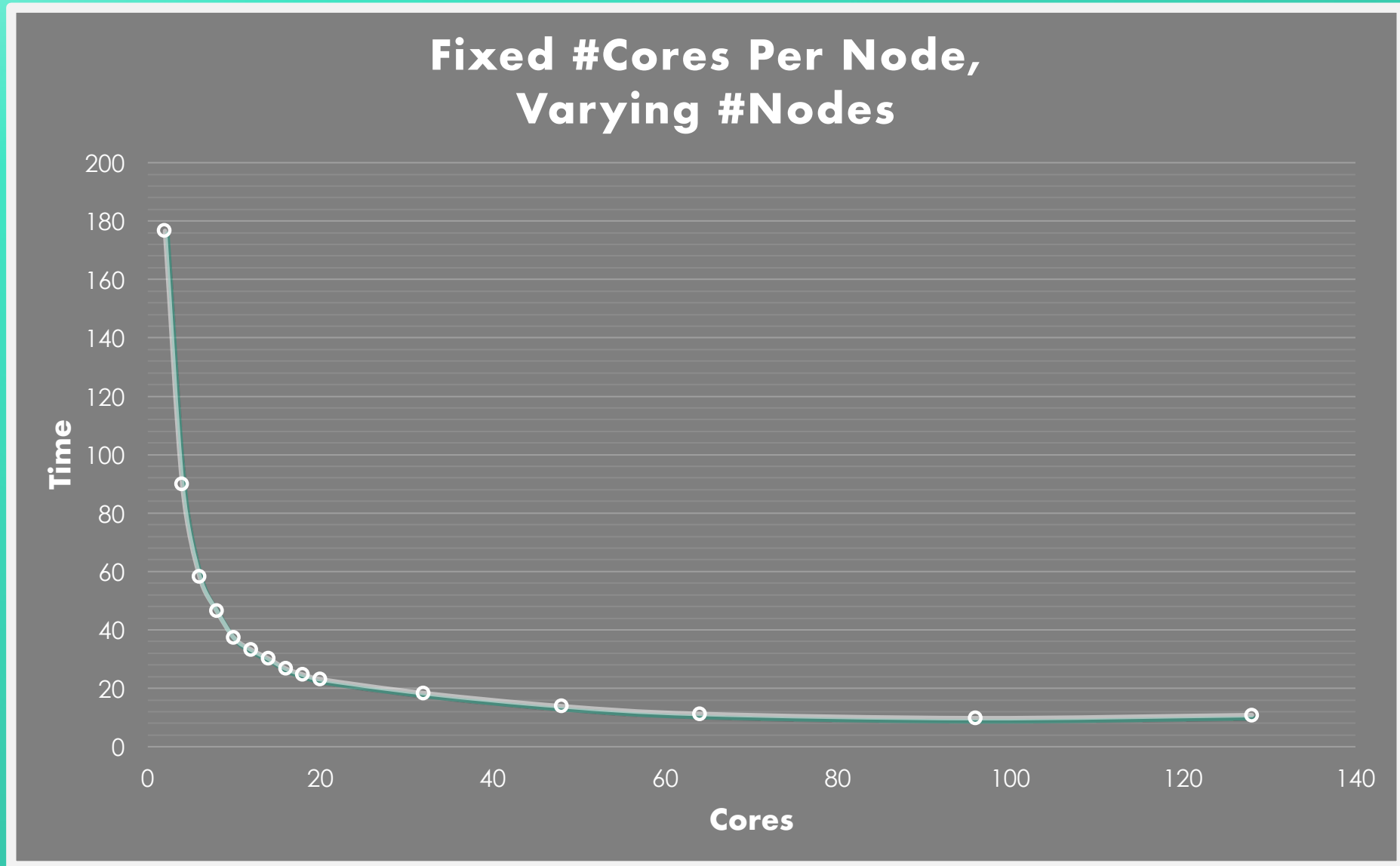
Results



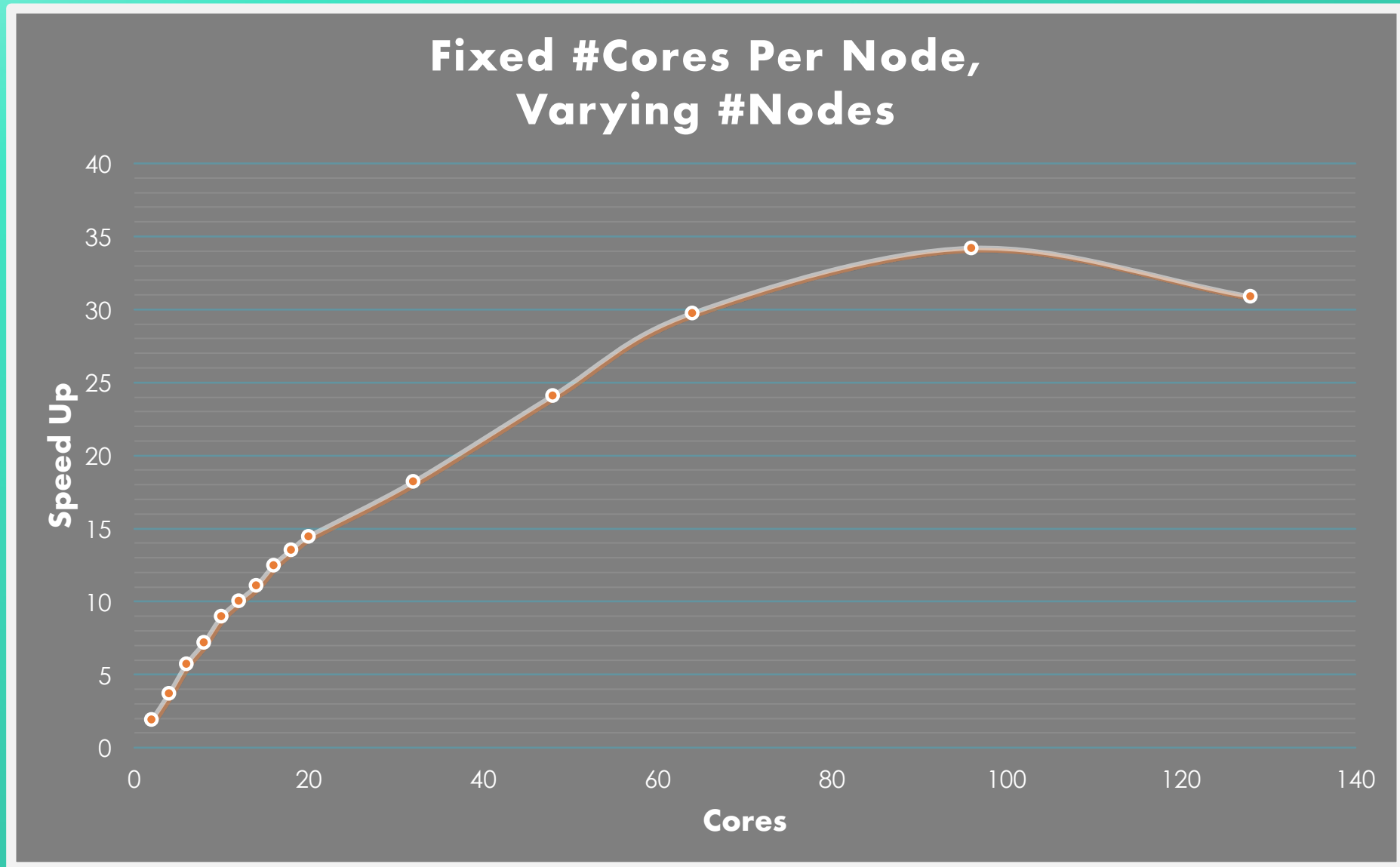
Results - Speed Up




Results



Results – Speed Up



Conclusion

- ▶ We can see that when the nodes are doubled, the time required to process/train the data decreases nearly by a factor of two. There is an additional overhead involved because of message passing.
 - ▶ As the data partitions become small, the message passing overhead dominates the processing time.
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

References

- [1] <http://whatis.techtarget.com/definition/machine-learning>
- [2] Miller, Russ, and Laurence Boxer. Algorithms Sequential & Parallel: A Unified Approach. 3rd ed., 2012.
- [3] <https://www.mathworks.com/products/distriben.html>
- [4] <https://en.wikipedia.org/wiki/MapReduce>
- [5] Dean J, Ghemawat S. 2008. MapReduce: Simplified data processing on large clusters. Commun ACM 51: 107–113.

THANK YOU

