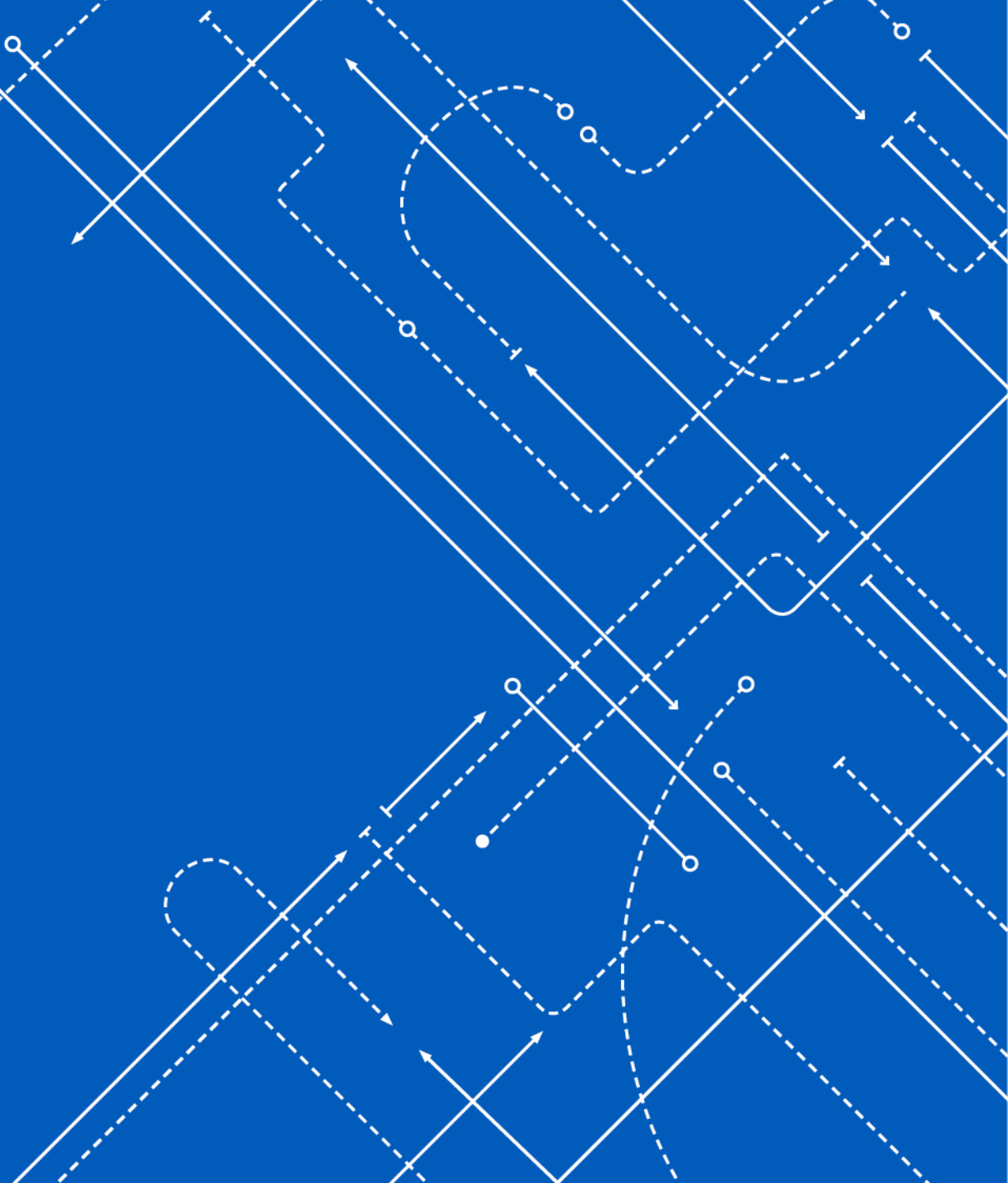
PARALLEL LOGISTIC REGRESSION

Varun Bhatt

CSE 633

University at Buffalo The State University of New York



Logistic Regression

- Cost Function $$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \times log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \times log(h_\theta(x^{(i)}))\right]$$
- Minimize Cost $$\min_{\theta} J(\theta)$$
- Gradient Descent $$\theta_j \leftarrow \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$
- Gradient Update Rule $$\theta_j \leftarrow \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

Equations credit: A. Aylin Tokuç “Gradient Descent Equation in Logistic Regression.” Baeldung

Gradient Descent: Sequential Algorithm

- `Initialize` θ `randomly`
- `for j in range(epochs):`

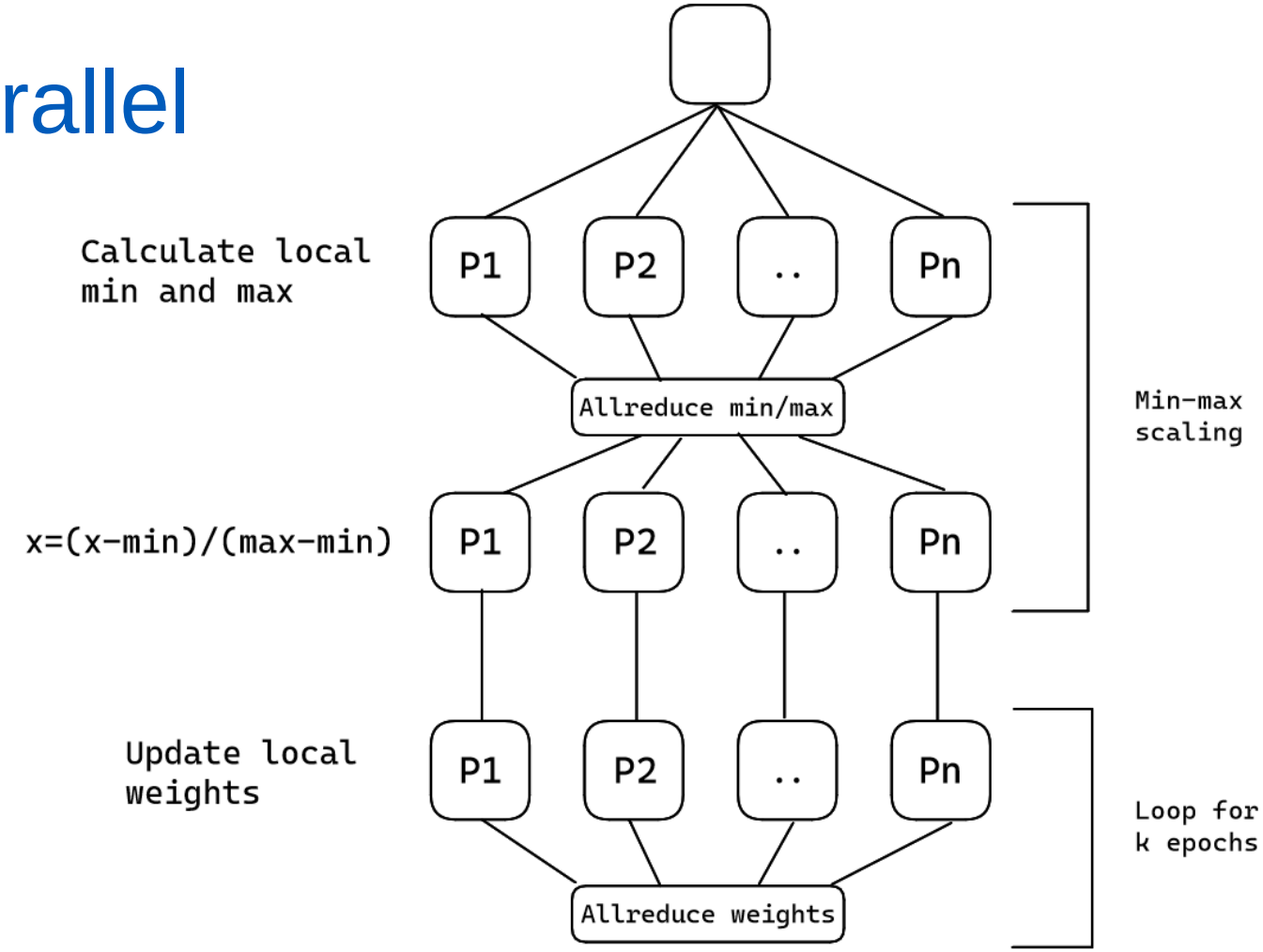
$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Gradient Descent in Parallel

- Divide input x across processors
- Initialize θ randomly
- Min-max scaling
- for j in range(epochs):

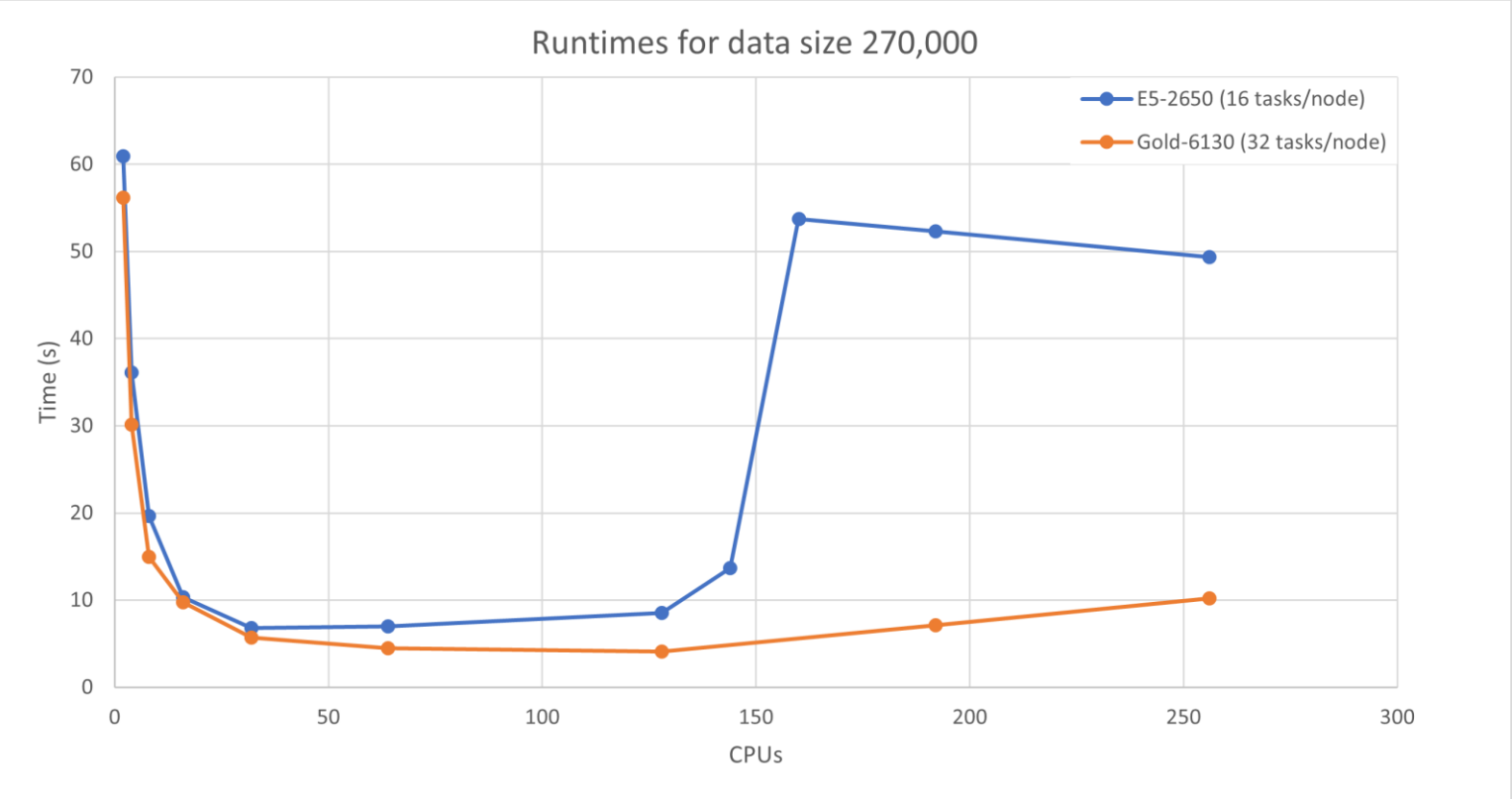
$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

  - Compute average of θ across processors



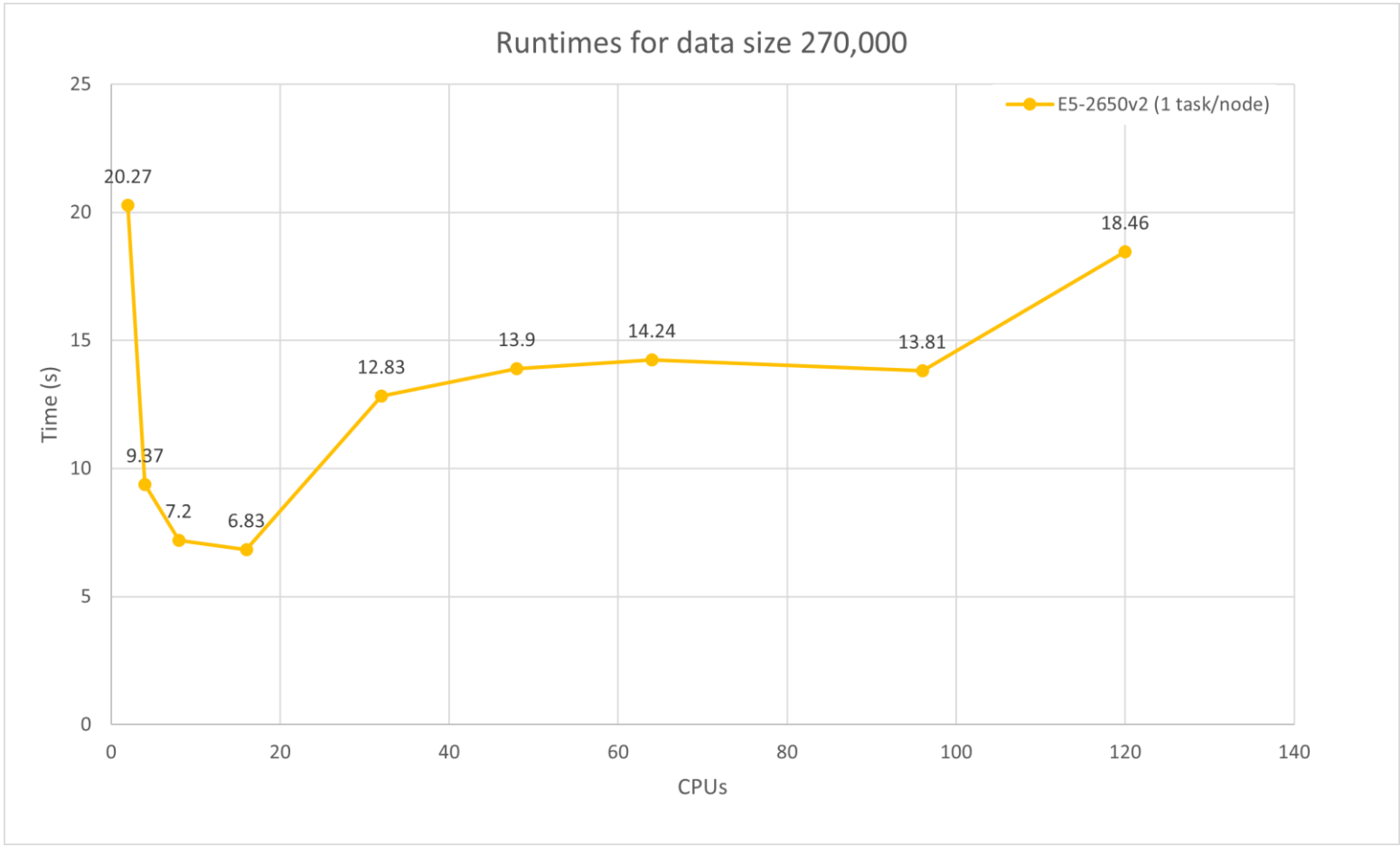
Results: Multiple tasks per node

	Time (s)	
Processors	E5-2650	Gold-6130
2	60.94	56.18
4	36.08	30.15
8	19.66	14.93
16	10.32	9.76
32	6.82	5.69
64	6.99	4.46
128	8.56	4.11
144	13.65	
160	53.71	
192	52.29	7.14
256	49.37	10.21



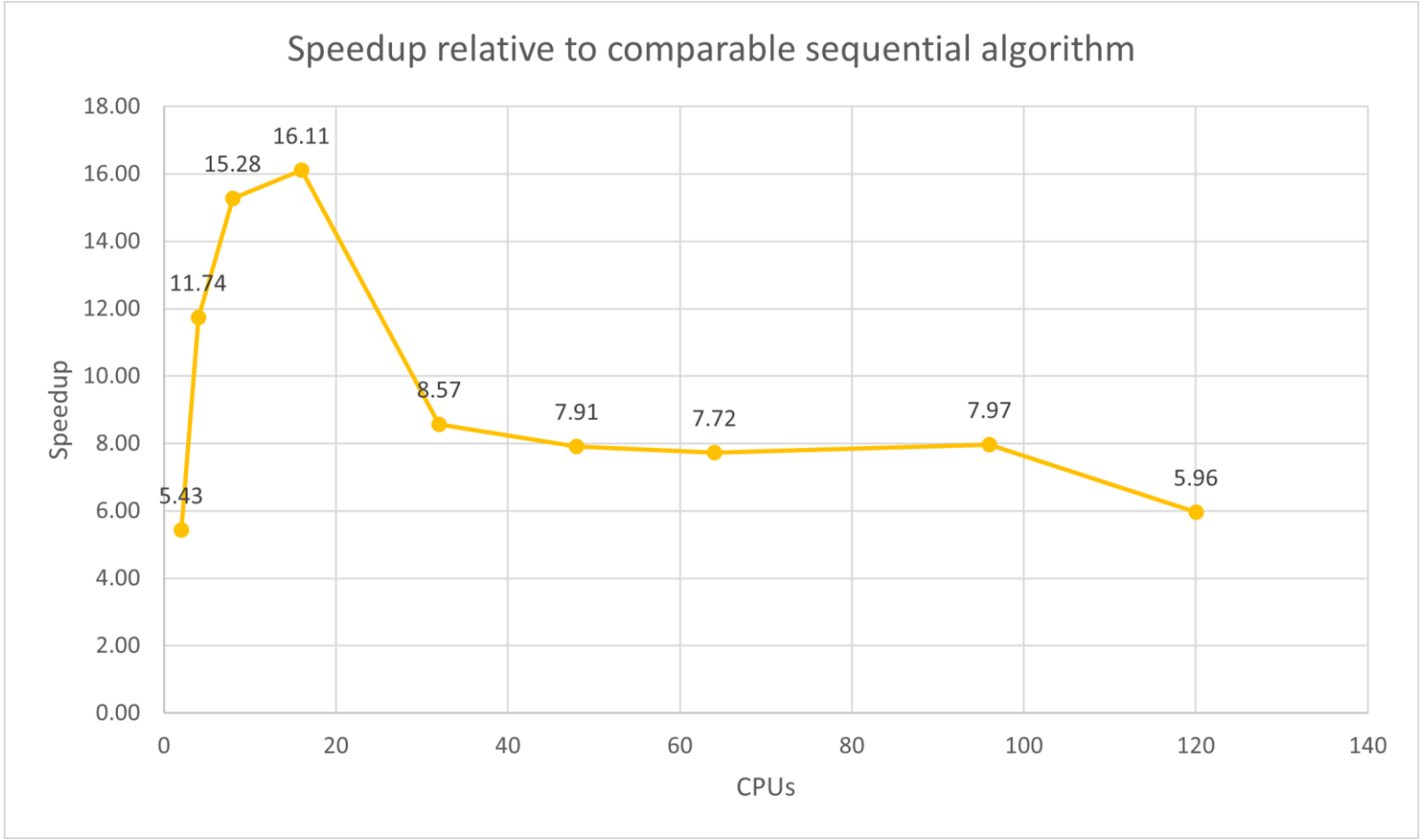
Results: 1 task per node

Processors	Time (s)
2	20.27
4	9.37
8	7.2
16	6.83
32	12.83
48	13.9
64	14.24
96	13.81
120	18.46



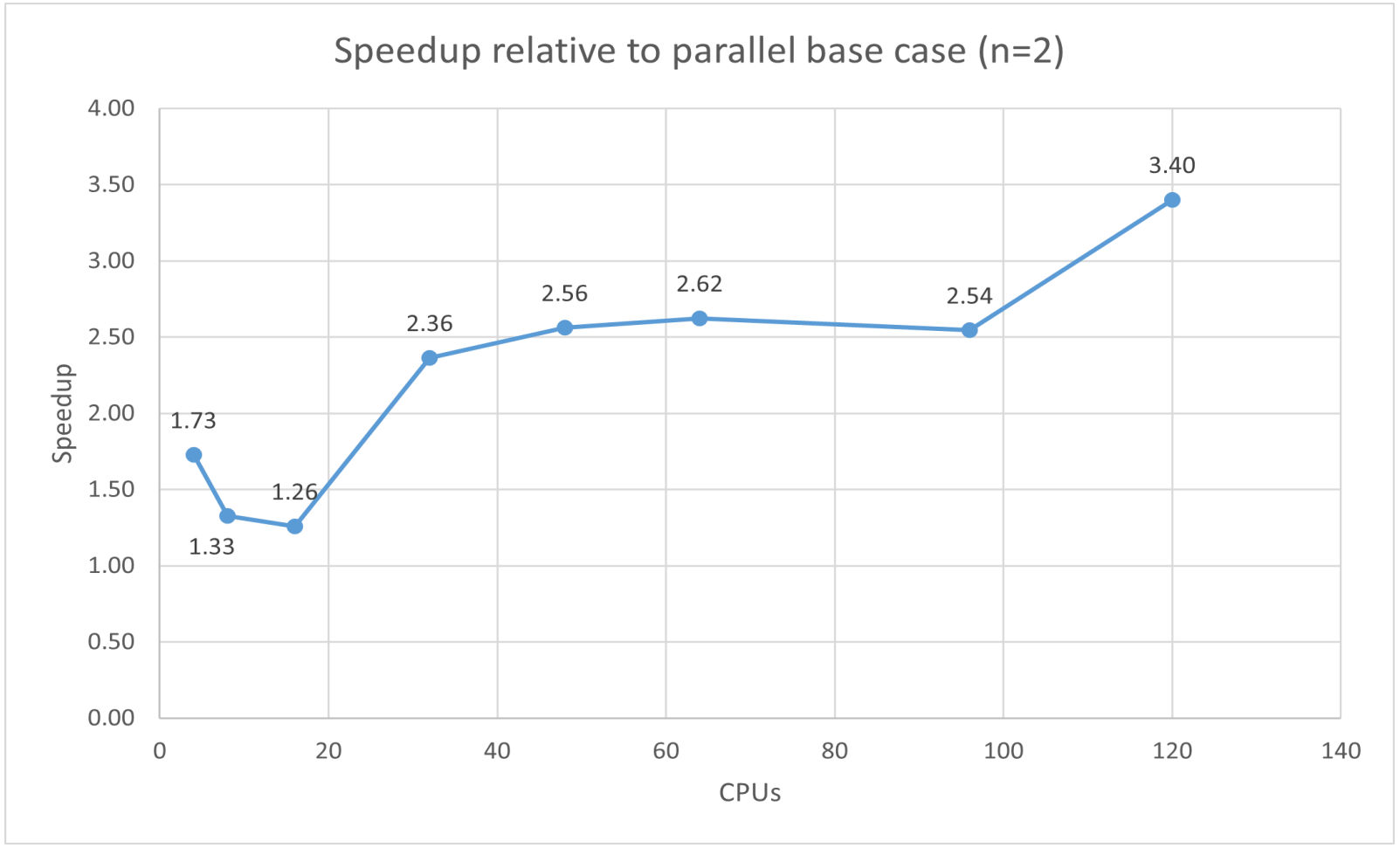
Results: Speedup

Processors	Speedup
2	5.43
4	11.74
8	15.28
16	16.11
32	8.57
48	7.91
64	7.72
96	7.97
120	5.96



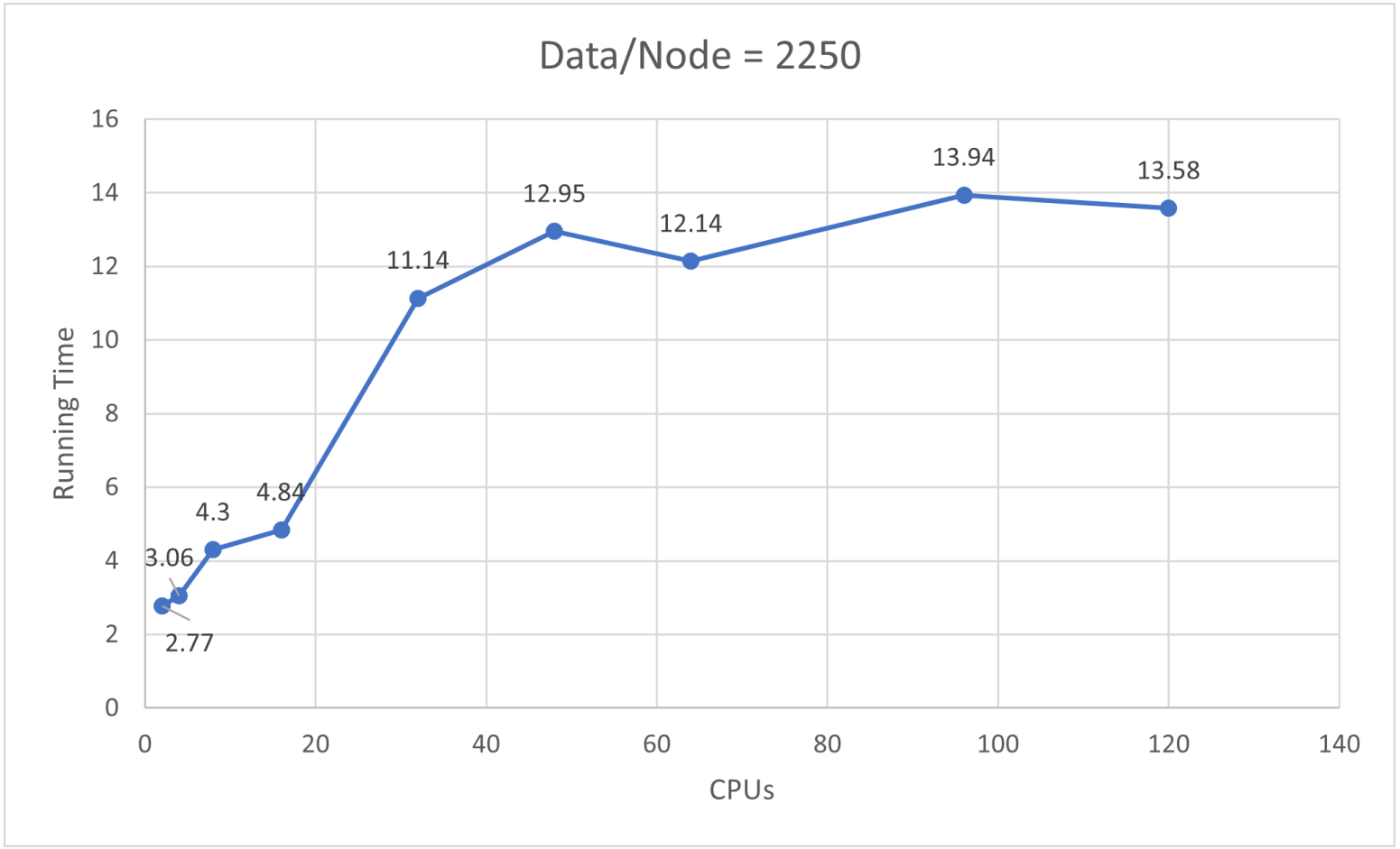
Results: Speedup

Processors	Speedup
2	
4	1.73
8	1.33
16	1.26
32	2.36
48	2.56
64	2.62
96	2.54
120	3.40



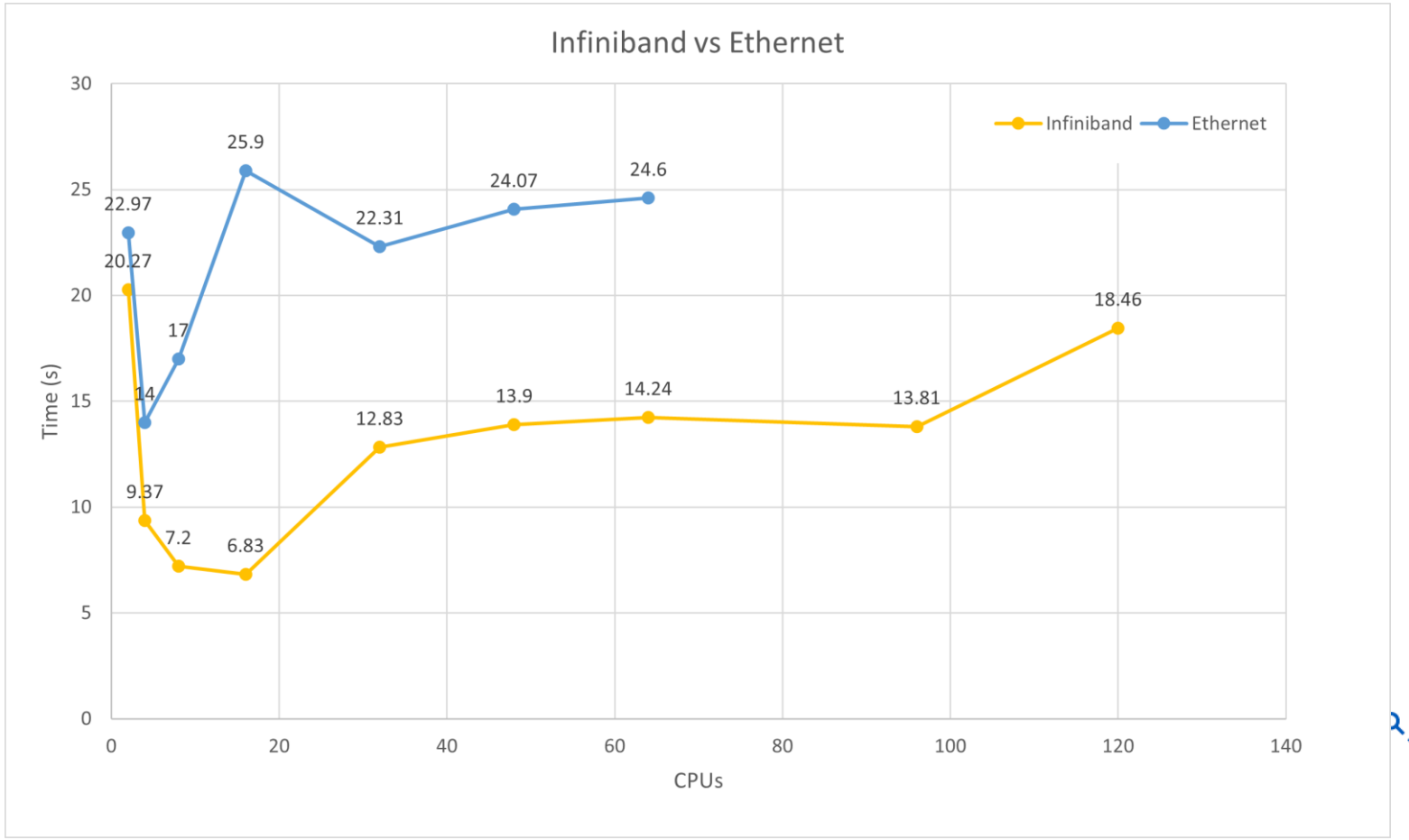
Results: Fixed size data per node

Processors	Speedup
2	2.77
4	3.06
8	4.3
16	4.84
32	11.14
48	12.95
64	12.14
96	13.94
120	13.58



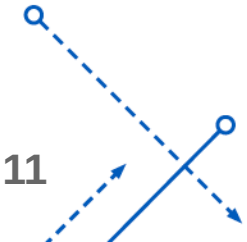
Results: Infiniband vs Ethernet

Processors	Infiniband	Ethernet
2	20.27	22.97
4	9.37	14
8	7.2	17
16	6.83	25.9
32	12.83	22.31
48	13.9	24.07
64	14.24	24.6
96	13.81	
120	18.46	



Future Work

- Try Stochastic Gradient Descent
- Explore Hogwild! algorithm



References

- A. Aylin Tokuç. “Gradient Descent Equation in Logistic Regression.” *Baeldung*, 11 Nov. 2022, https://www.baeldung.com/cs/gradient-descent-logistic-regression.
- default of credit card clients. (2016). UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H.