Faster Jobs in Distributed Data Processing using Multi-Task Learning

Neeraja J. Yadwadkar, Bharath Hariharan
Joseph Gonzalez and Randy Katz

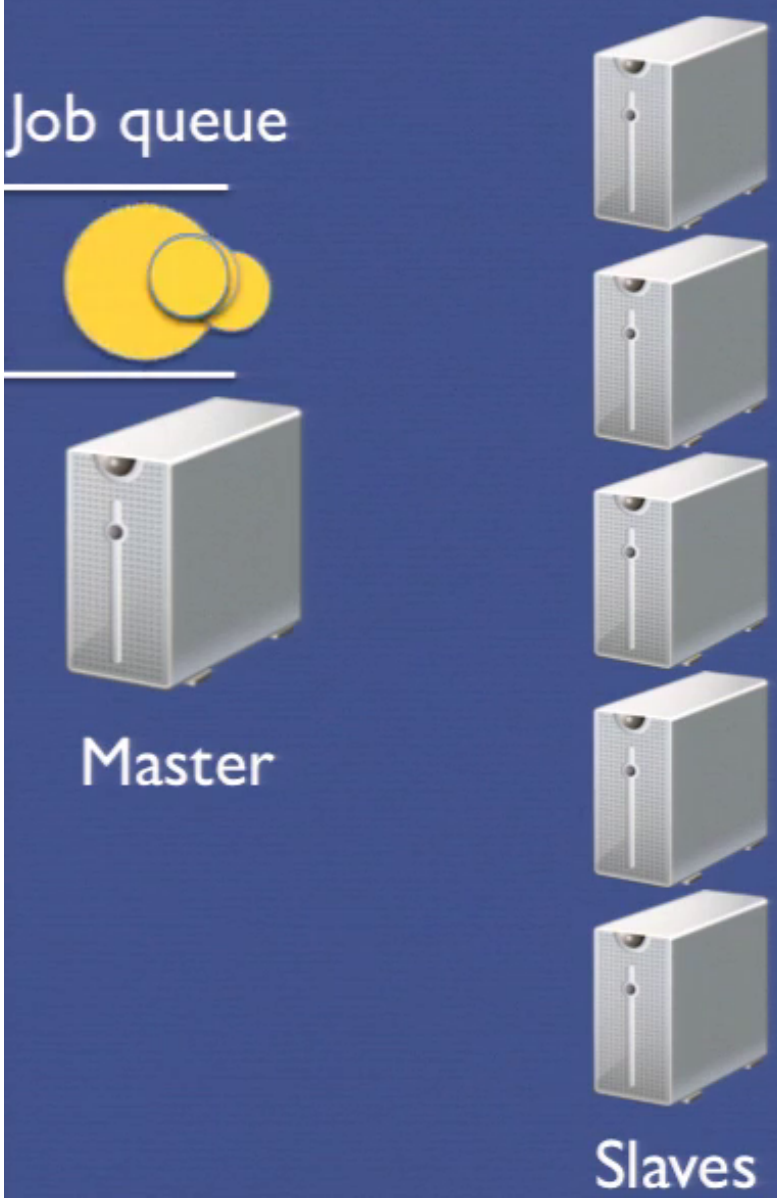
Parallel Data Analytics

Job queue

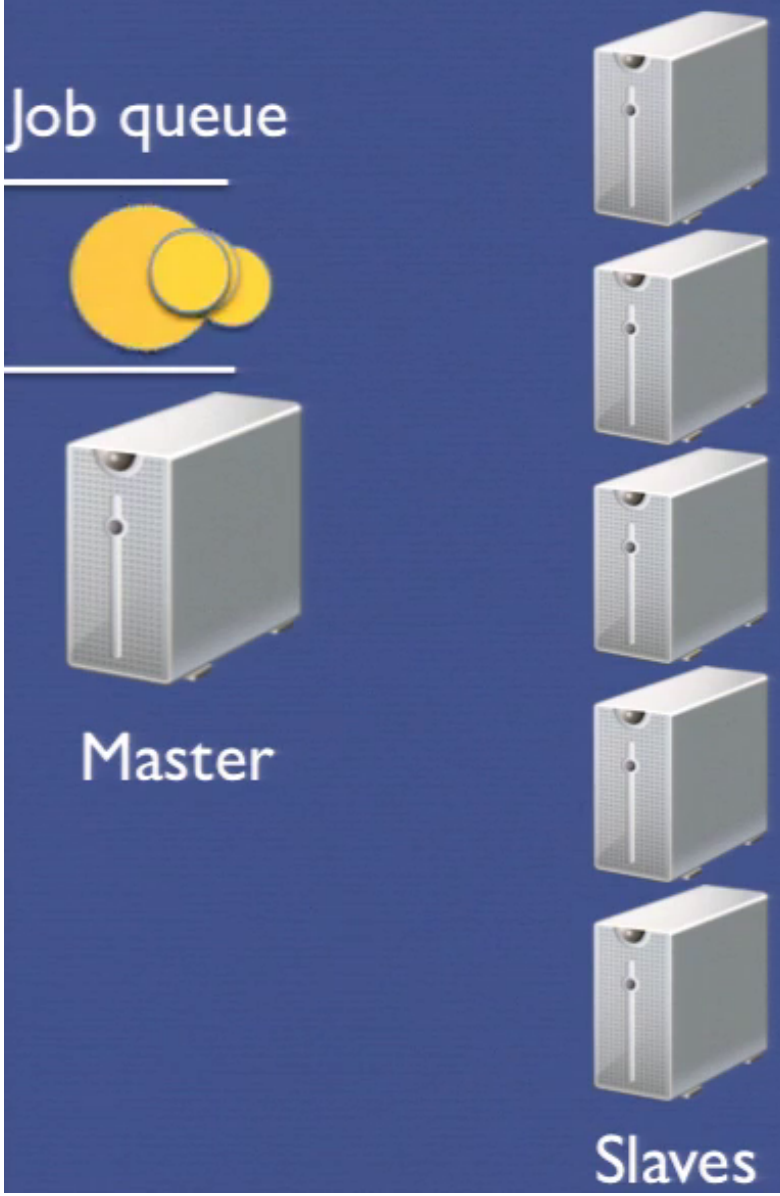
Master

Slaves

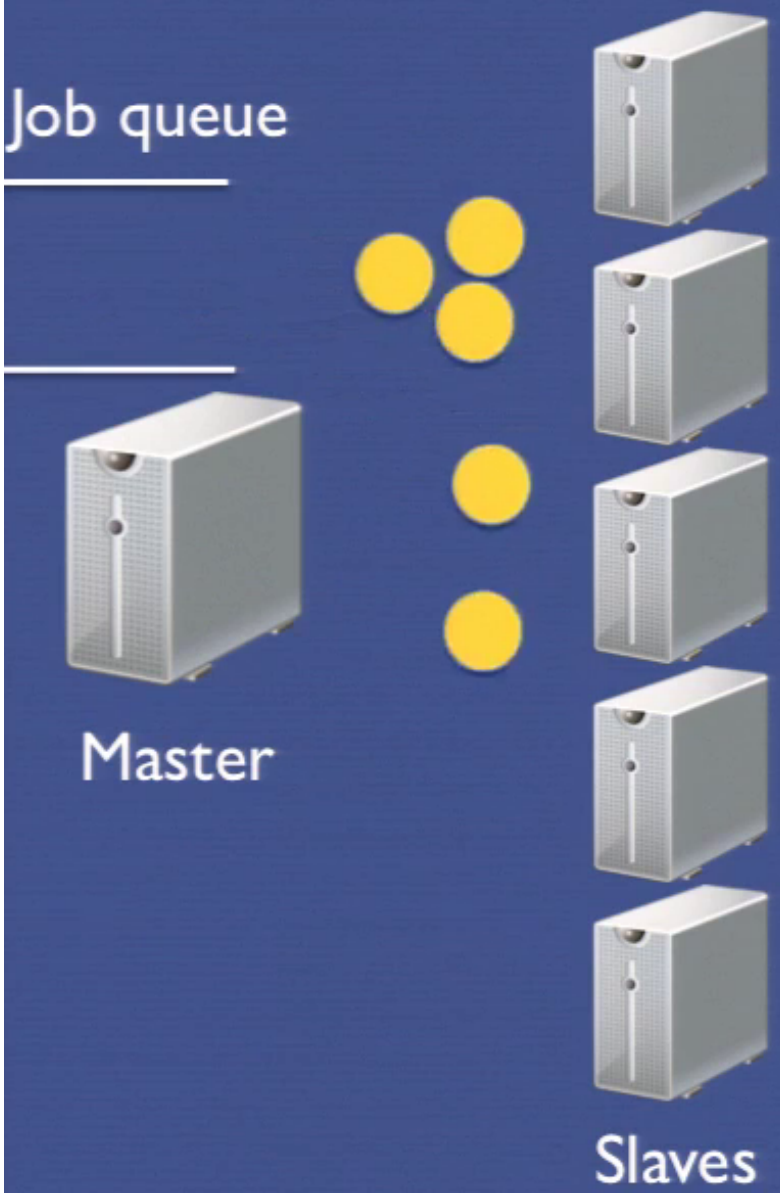
Parallel Data Analytics



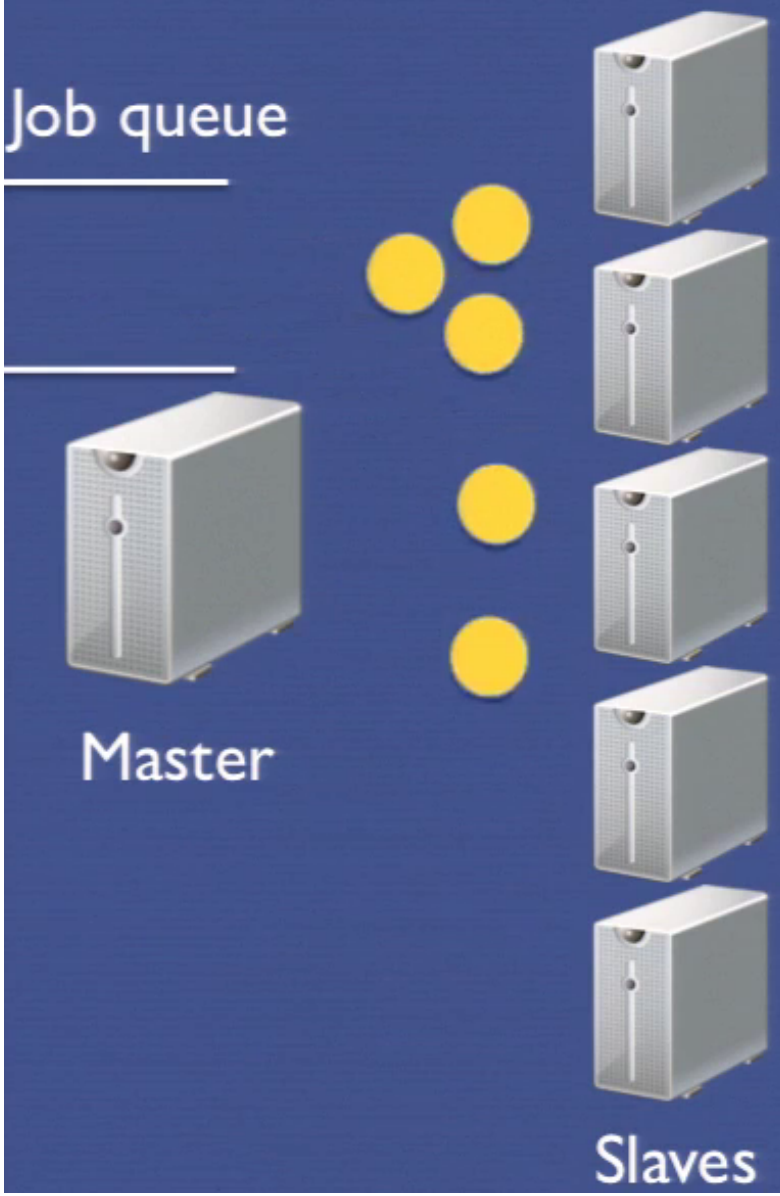
Parallel Data Analytics



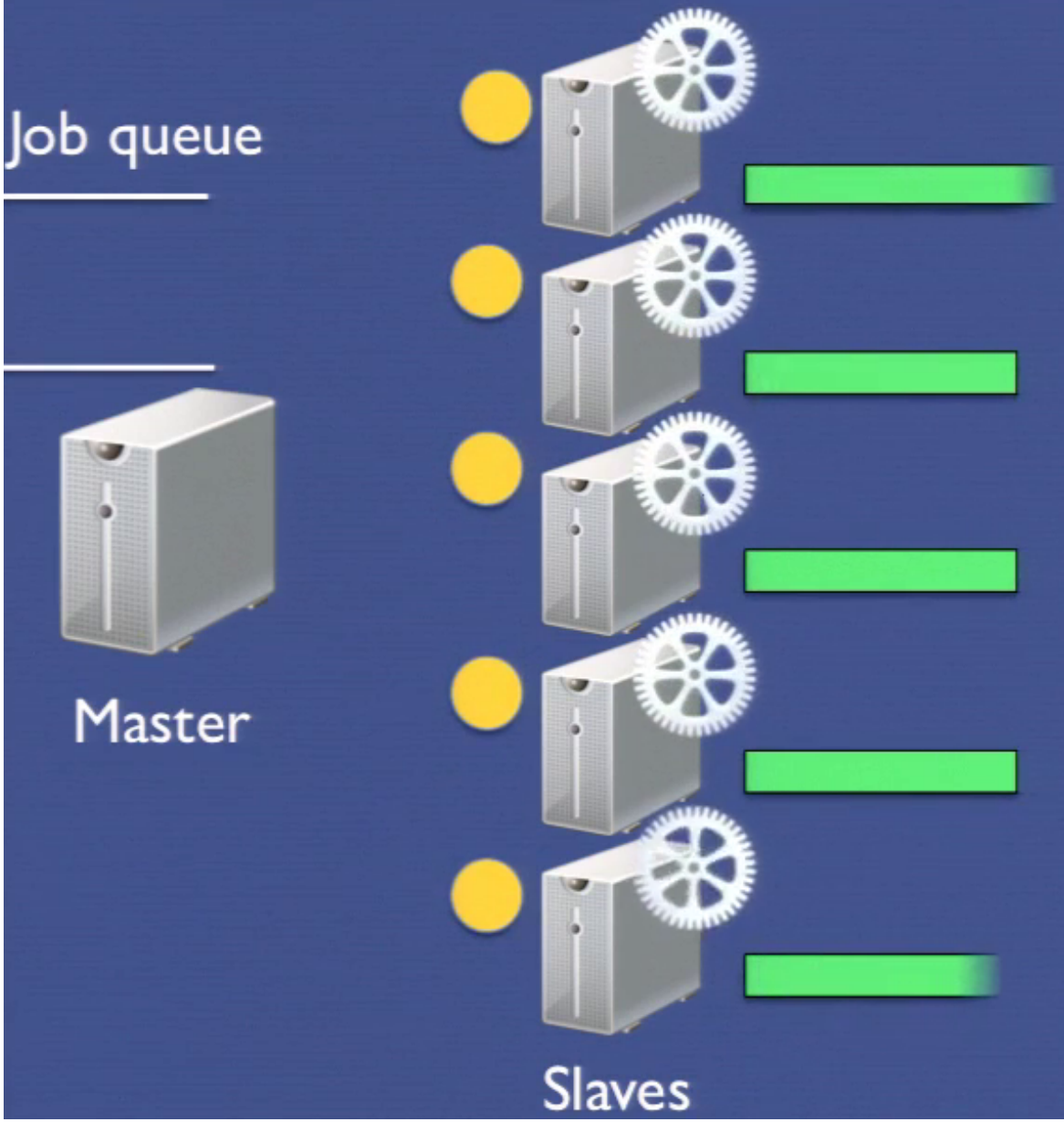
Parallel Data Analytics



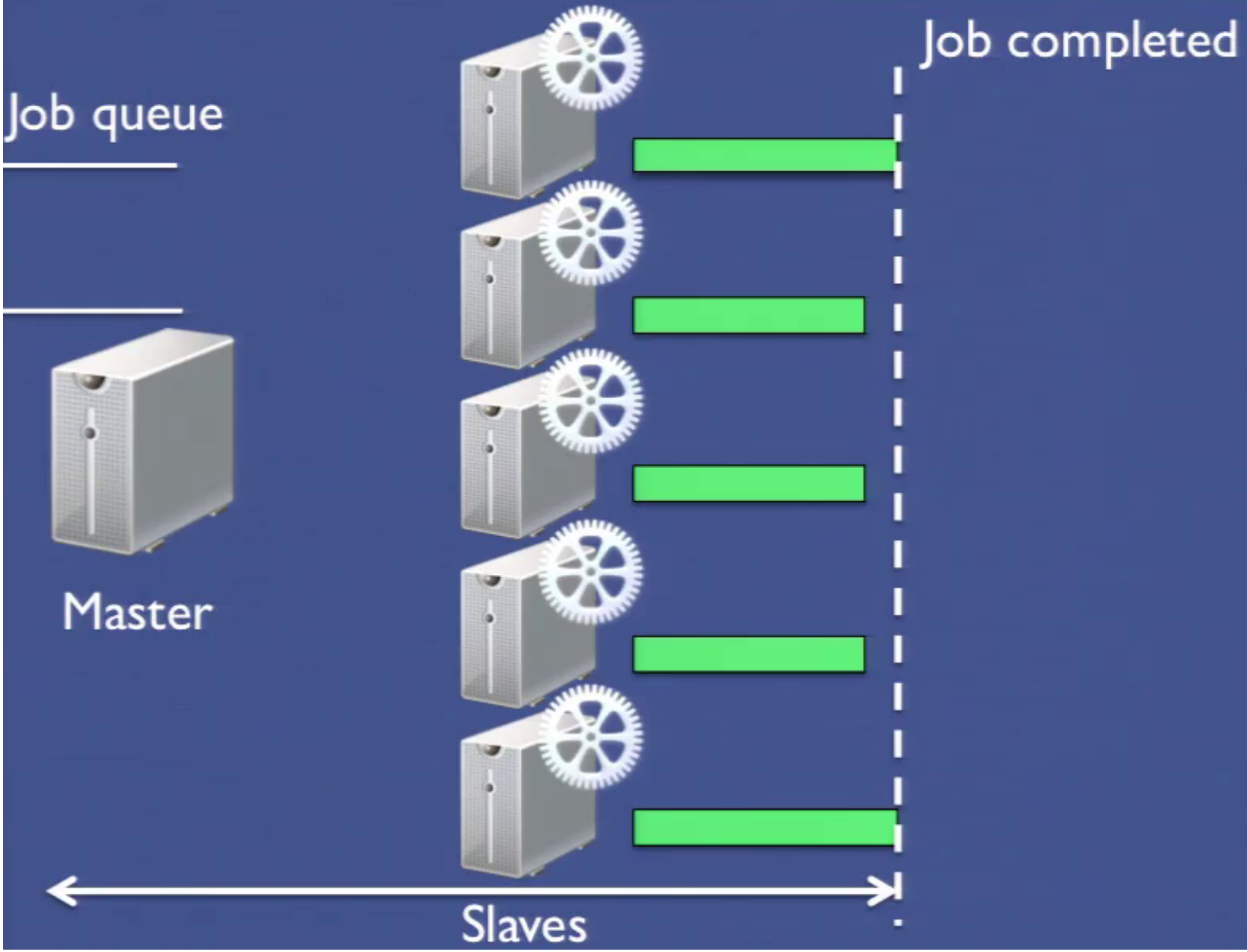
Parallel Data Analytics



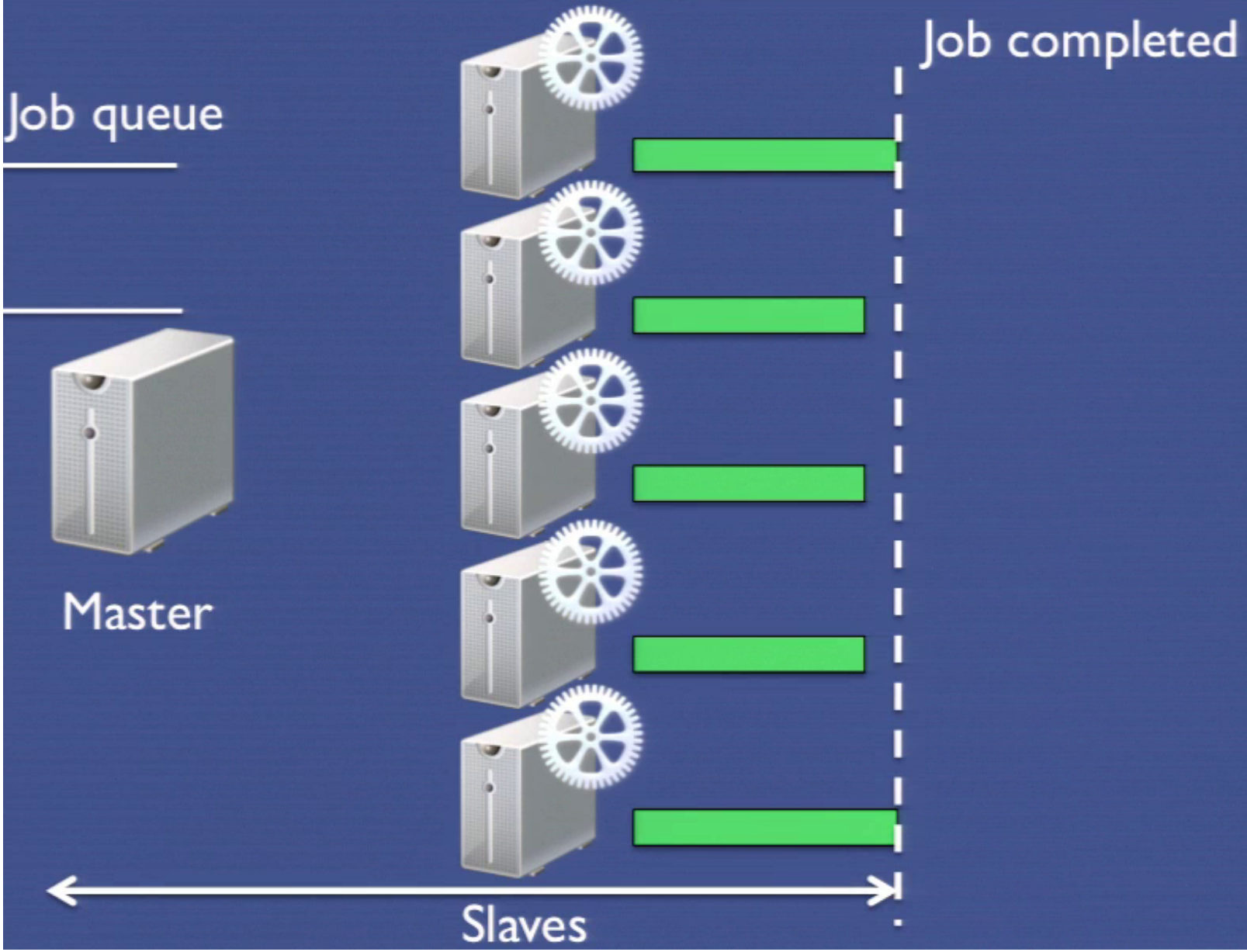
Parallel Data Analytics



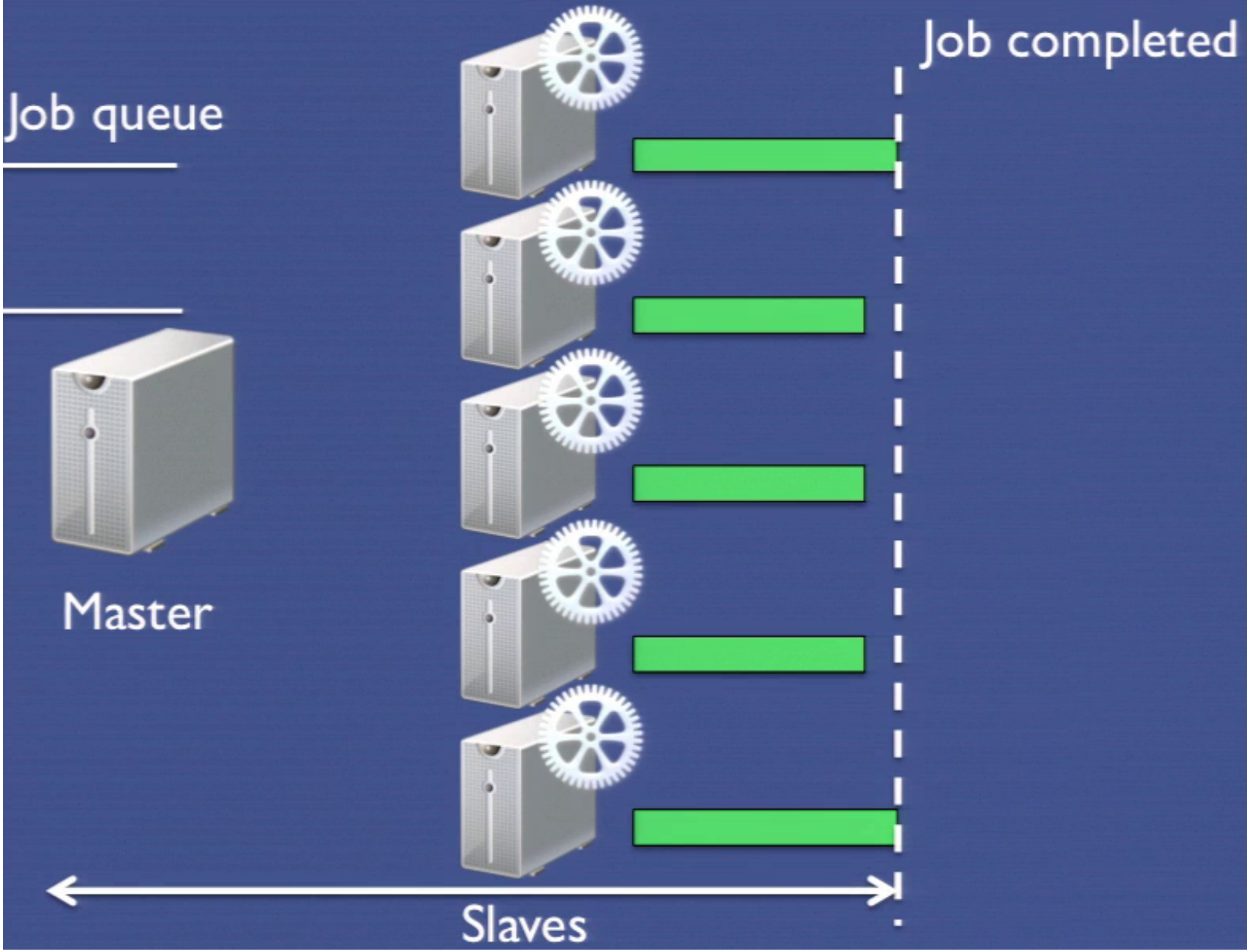
Parallel Data Analytics



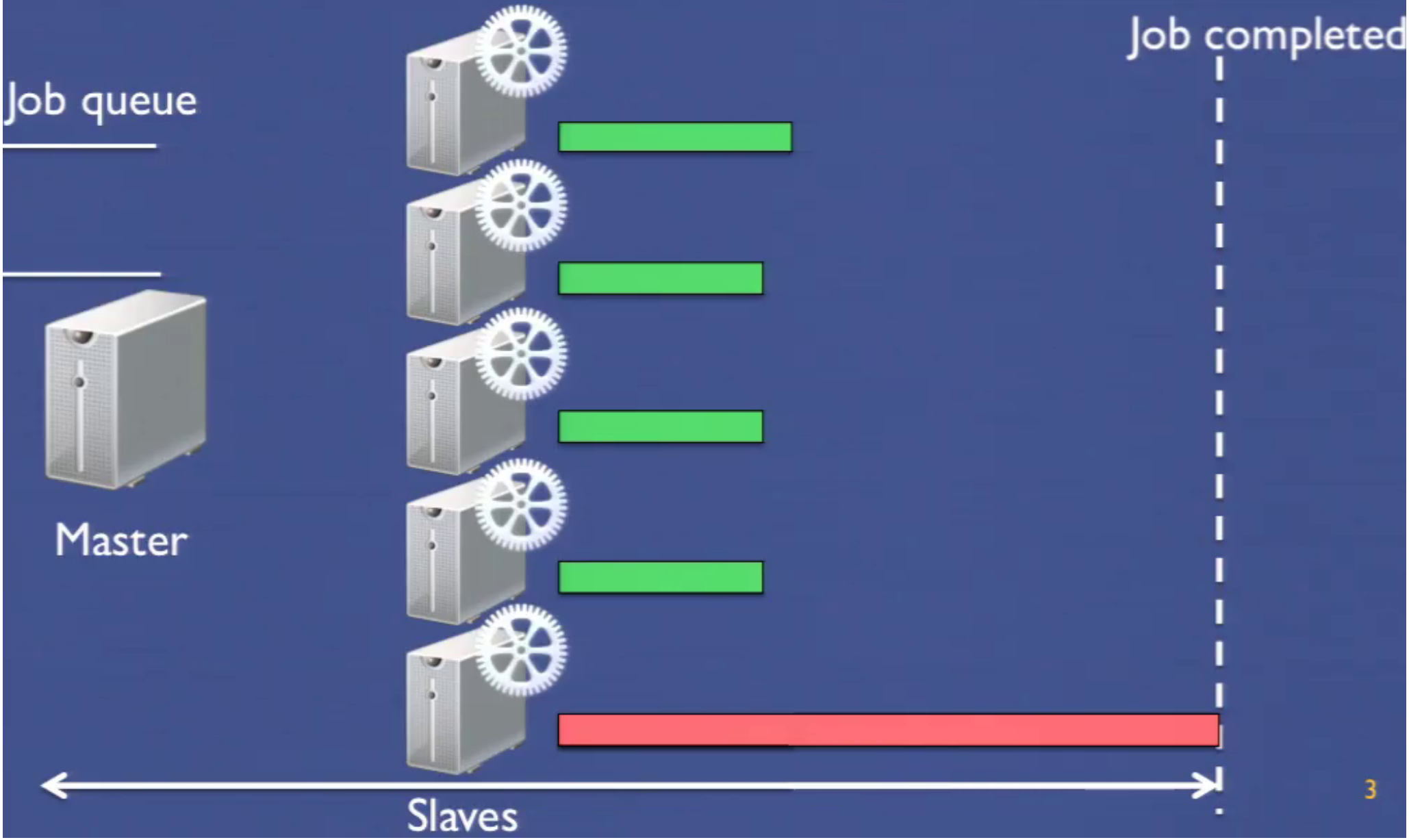
Stragglers



Stragglers



Stragglers



Defining Stragglers

Tasks of a job

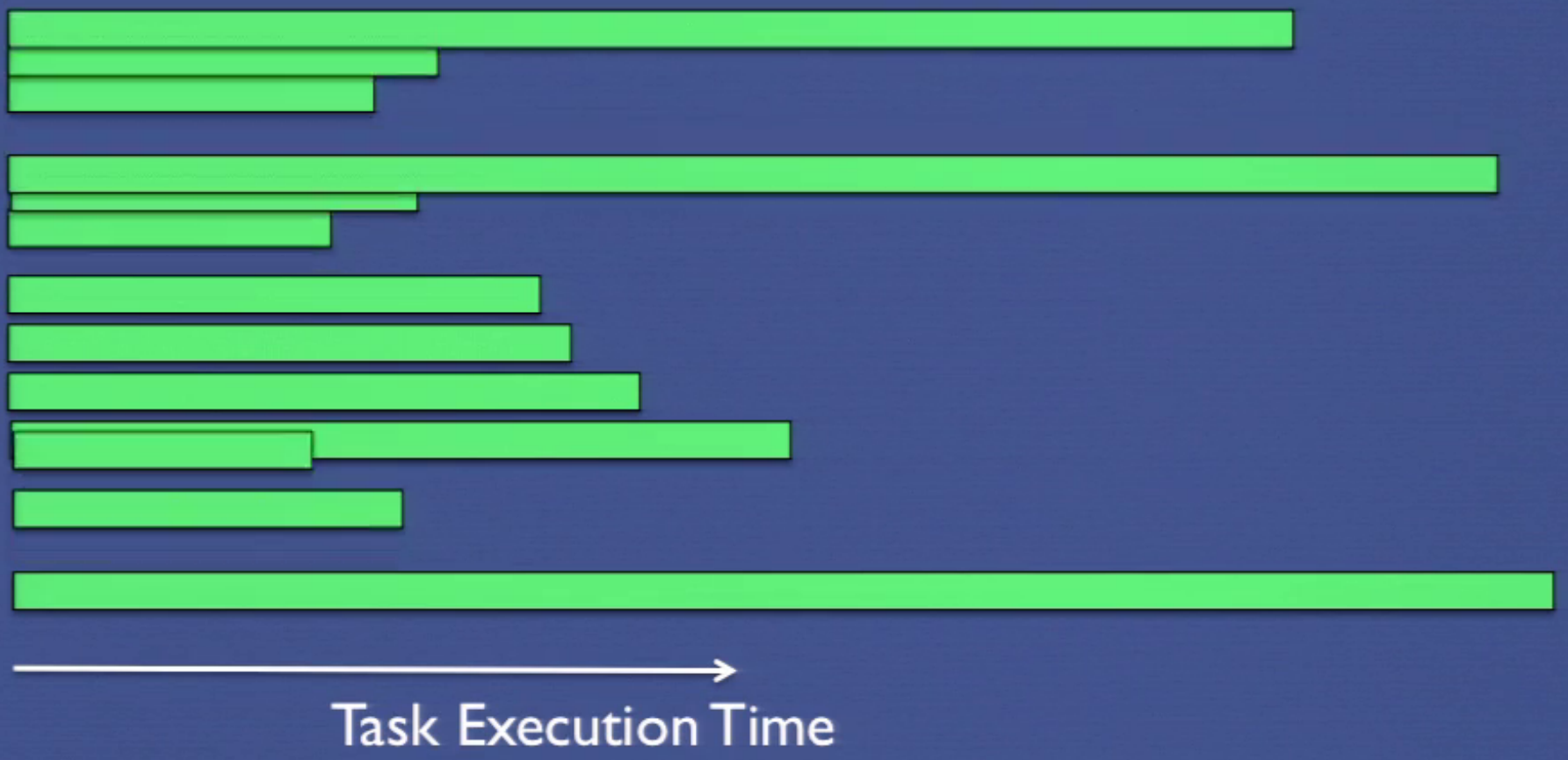
Task Execution Time

Defining Stragglers

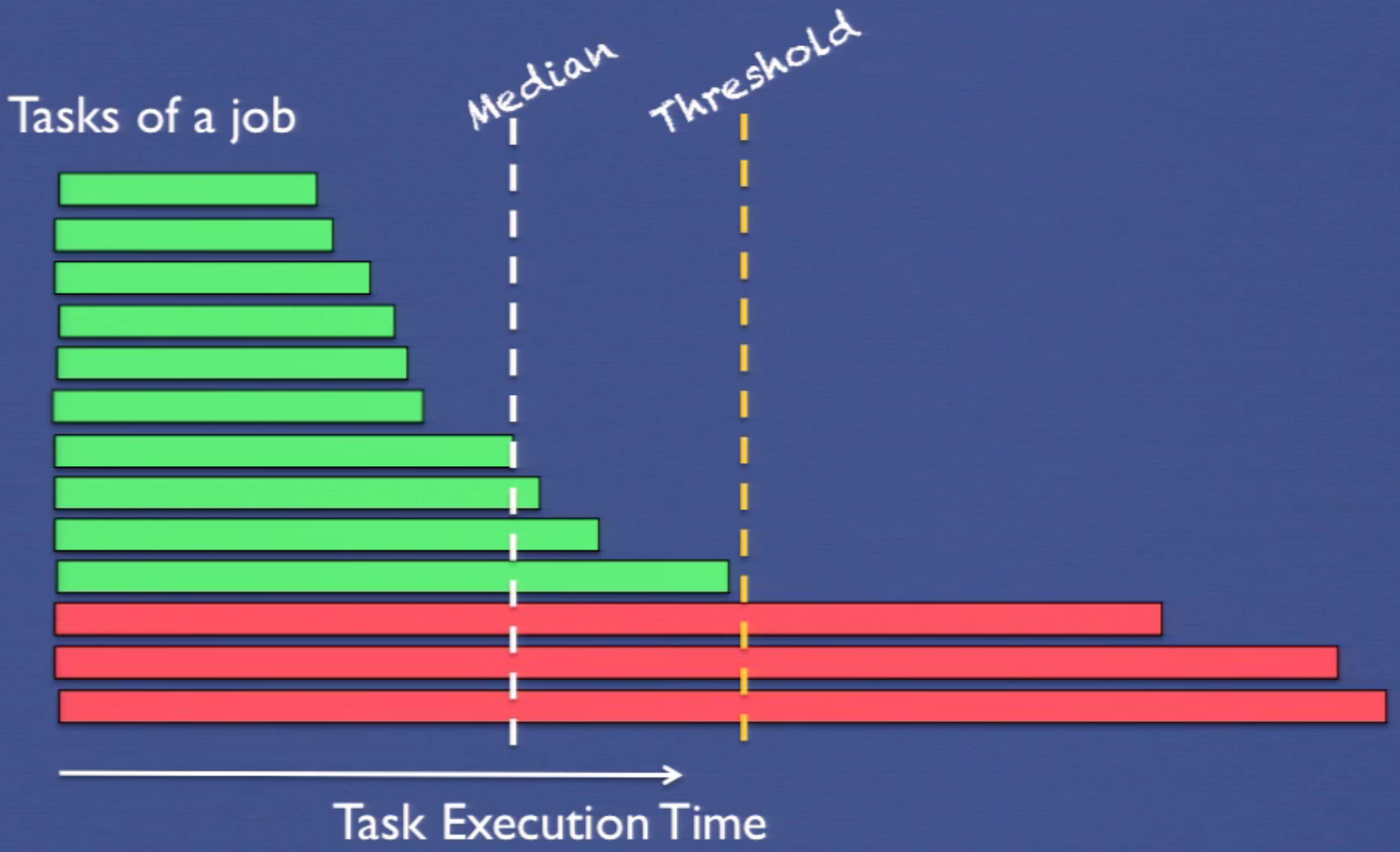
Tasks of a job

Defining Stragglers

Tasks of a job



Defining Stragglers



Impact of Stragglers

Presence of Stragglers in real-world production level traces*:

*captured for over 6 months from about 4000 machines in total

Impact of Stragglers

Presence of Stragglers in real-world production level traces*:

When replayed using SWIM+ on a 50 node EC2 cluster….

Workload	Stragglers
Facebook 2009 (FB2009)	
Facebook 2010 (FB2010)	
Cloudera's Customer b (CC_b)	
Cloudera's Customer e (CC_e)	

*captured for over 6 months from about 4000 machines in total

+Chen Y., et al., The Case for Evaluating MapReduce Performance Using Workload Suites, MASCOTS'11

Impact of Stragglers

Presence of Stragglers in real-world production level traces*:

When replayed using SWIM+ on a 50 node EC2 cluster....

Workload	Stragglers
Facebook 2009 (FB2009)	24 %
Facebook 2010 (FB2010)	23 %
Cloudera's Customer b (CC_b)	28 %
Cloudera's Customer e (CC_e)	22 %

Threshold = 1.3 * median

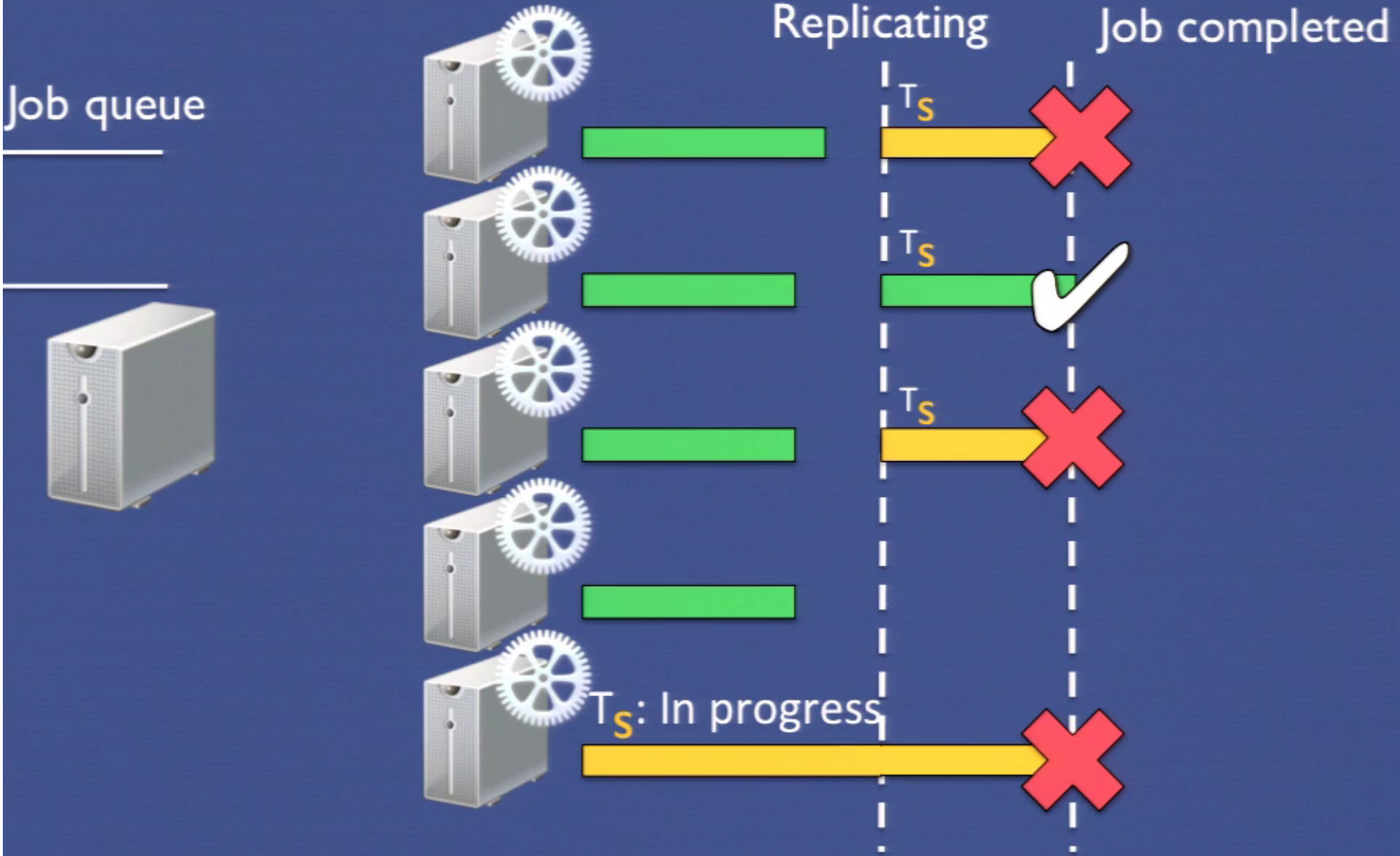
*captured for over 6 months from about 4000 machines in total

+Chen Y, et al, The Case for Evaluating MapReduce Performance Using Workload Suites, MASCOTS'11

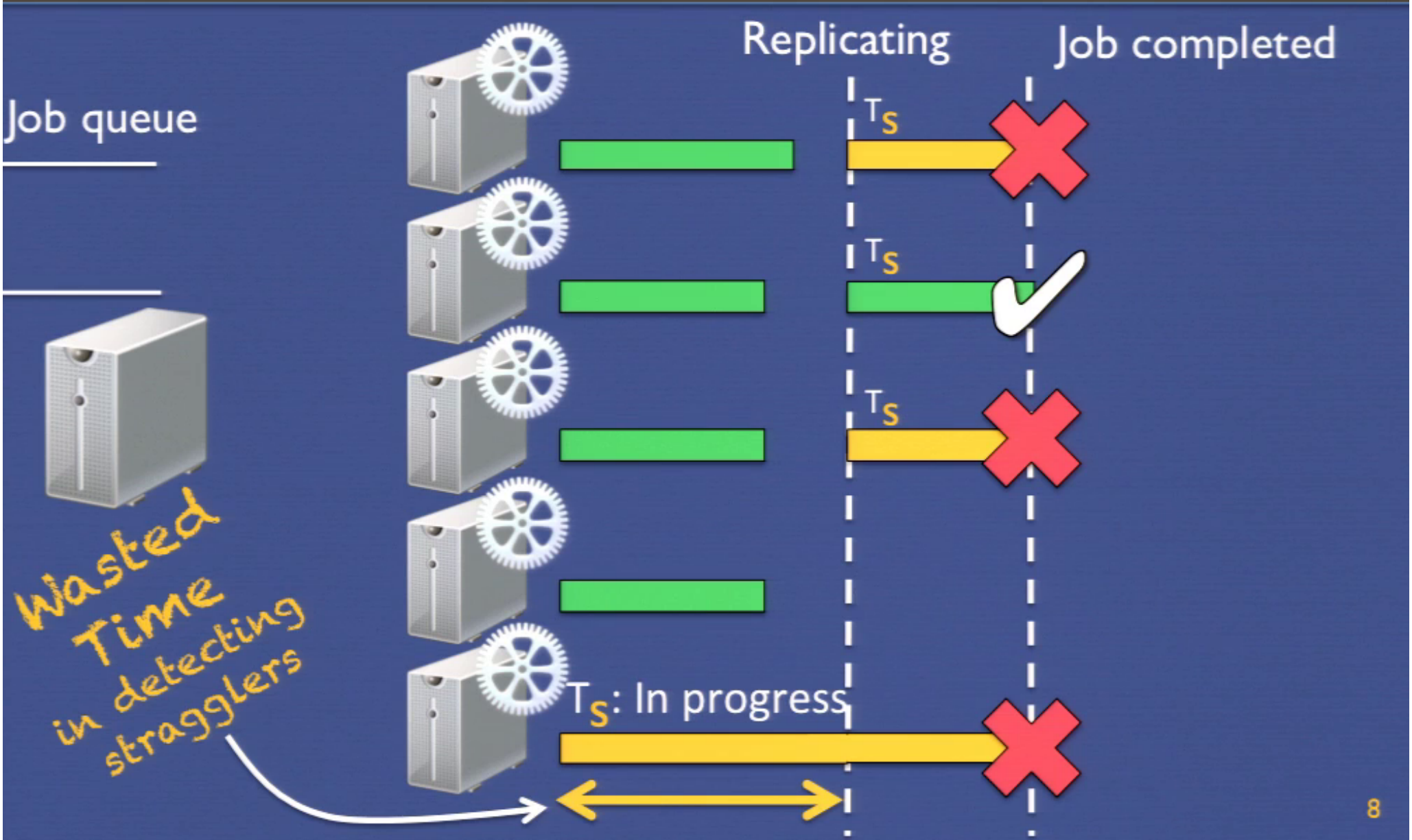
Outline

- ☑ Problem Context
- ❑ Existing Approaches:
  - ❑ Reactive: Speculative Execution
  - ❑ Proactive: Predictive modeling based approaches
- ❑ A New MTL Formulation
- ❑ Application to Straggler Avoidance
- ❑ Evaluation

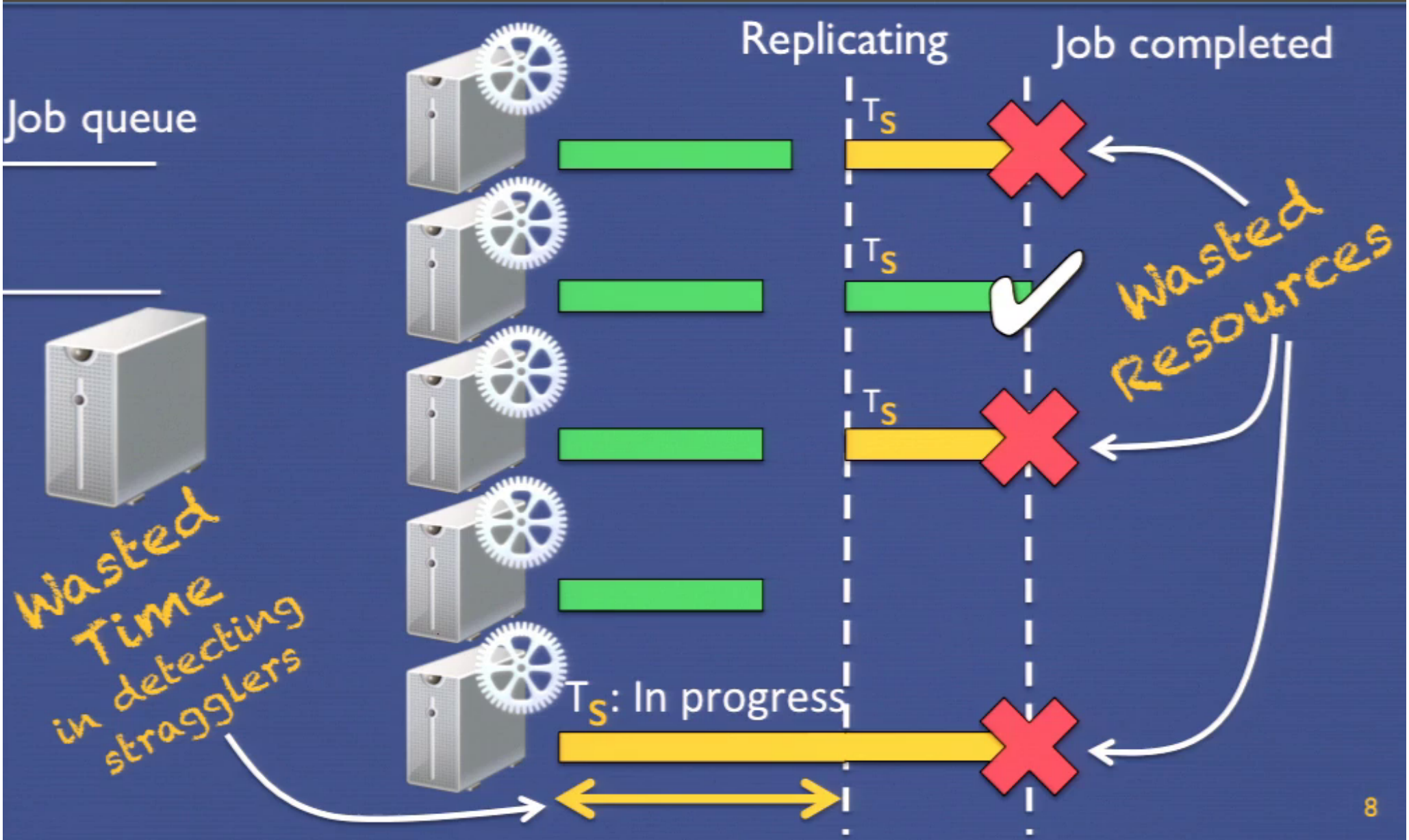
Existing *Reactive* Approach:
Speculative Execution



Existing *Reactive* Approach:
Speculative Execution



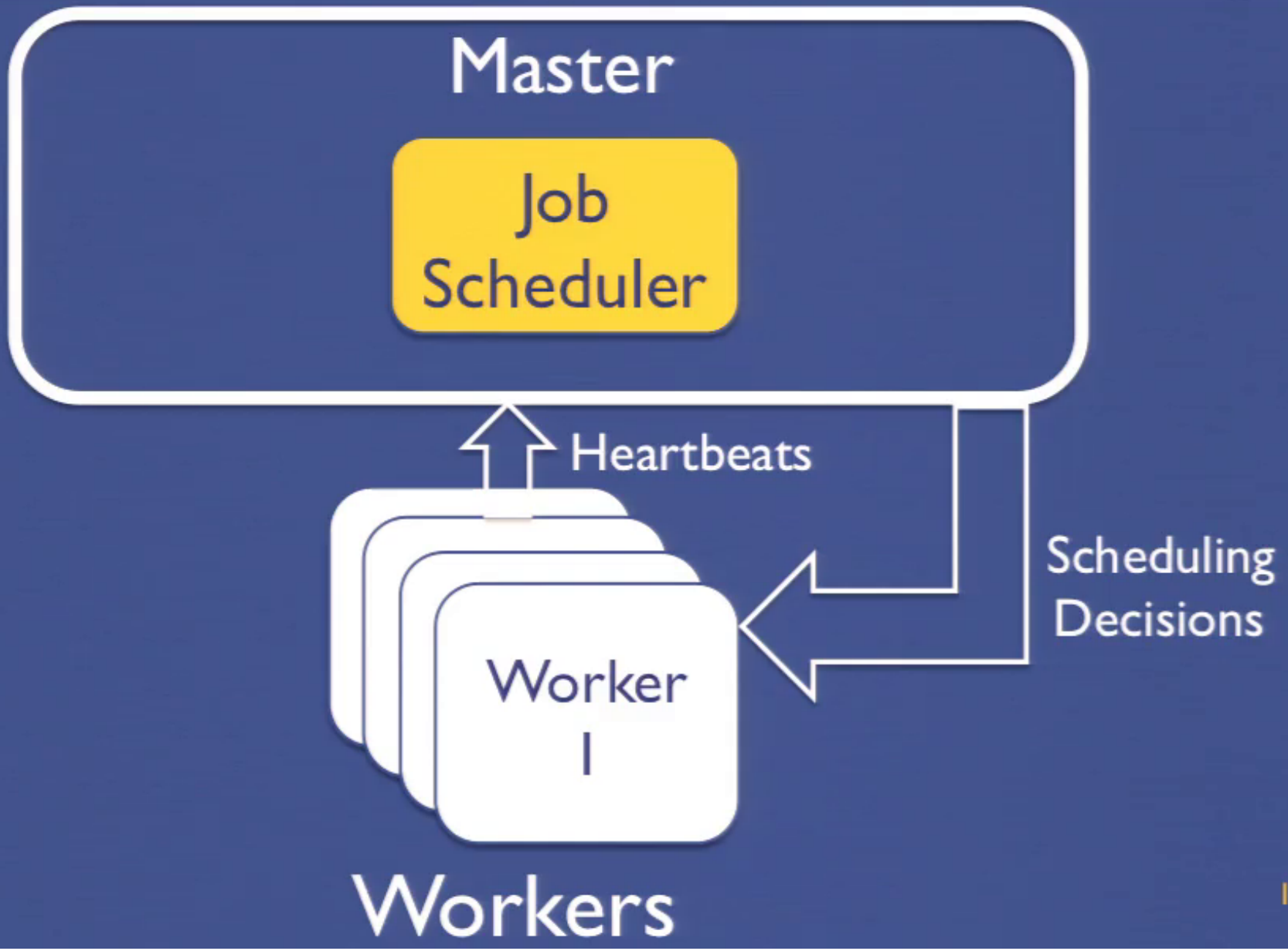
Existing *Reactive* Approach:
Speculative Execution



Outline

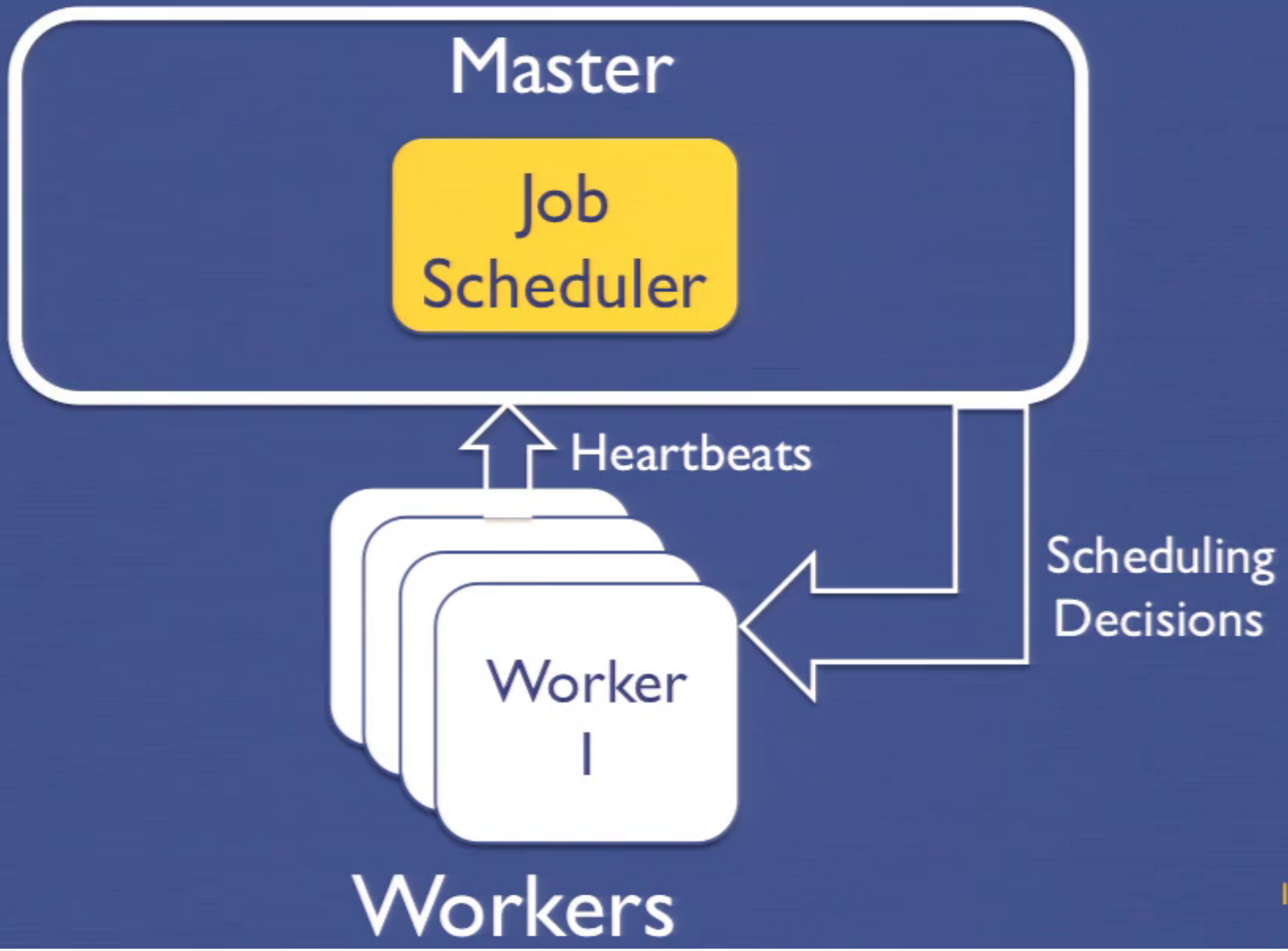
- ☑ Problem Context
- ☐ Existing Approaches:
  - ☑ Reactive: Speculative Execution
  - ☐ Proactive: Predictive modeling based approaches
- ☐ A New MTL Formulation
- ☐ Application to Straggler Avoidance
- ☐ Evaluation

Scheduling in MapReduce-based Frameworks



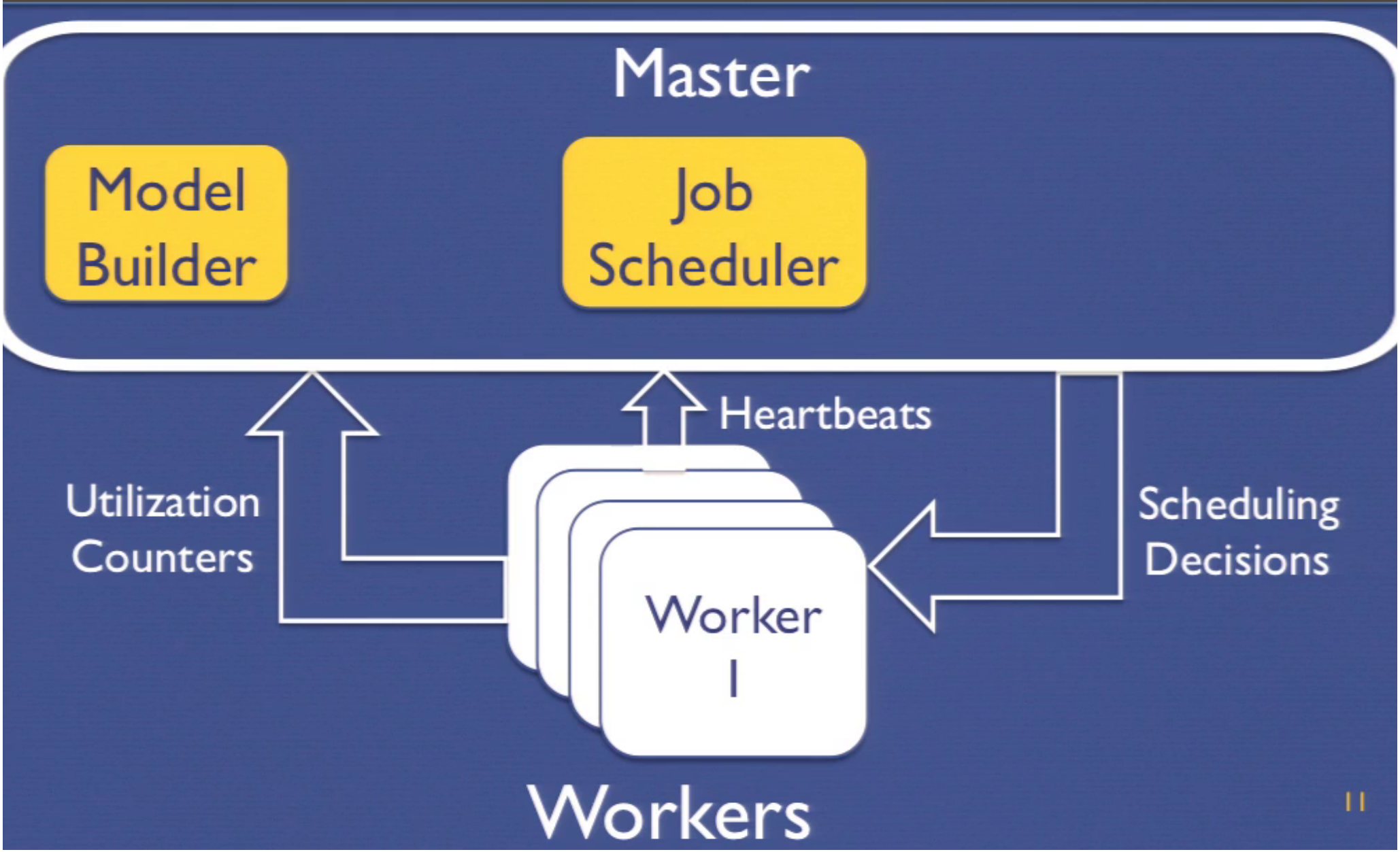
Existing *Proactive* Approach: Wrangler

Predict stragglers to avoid them



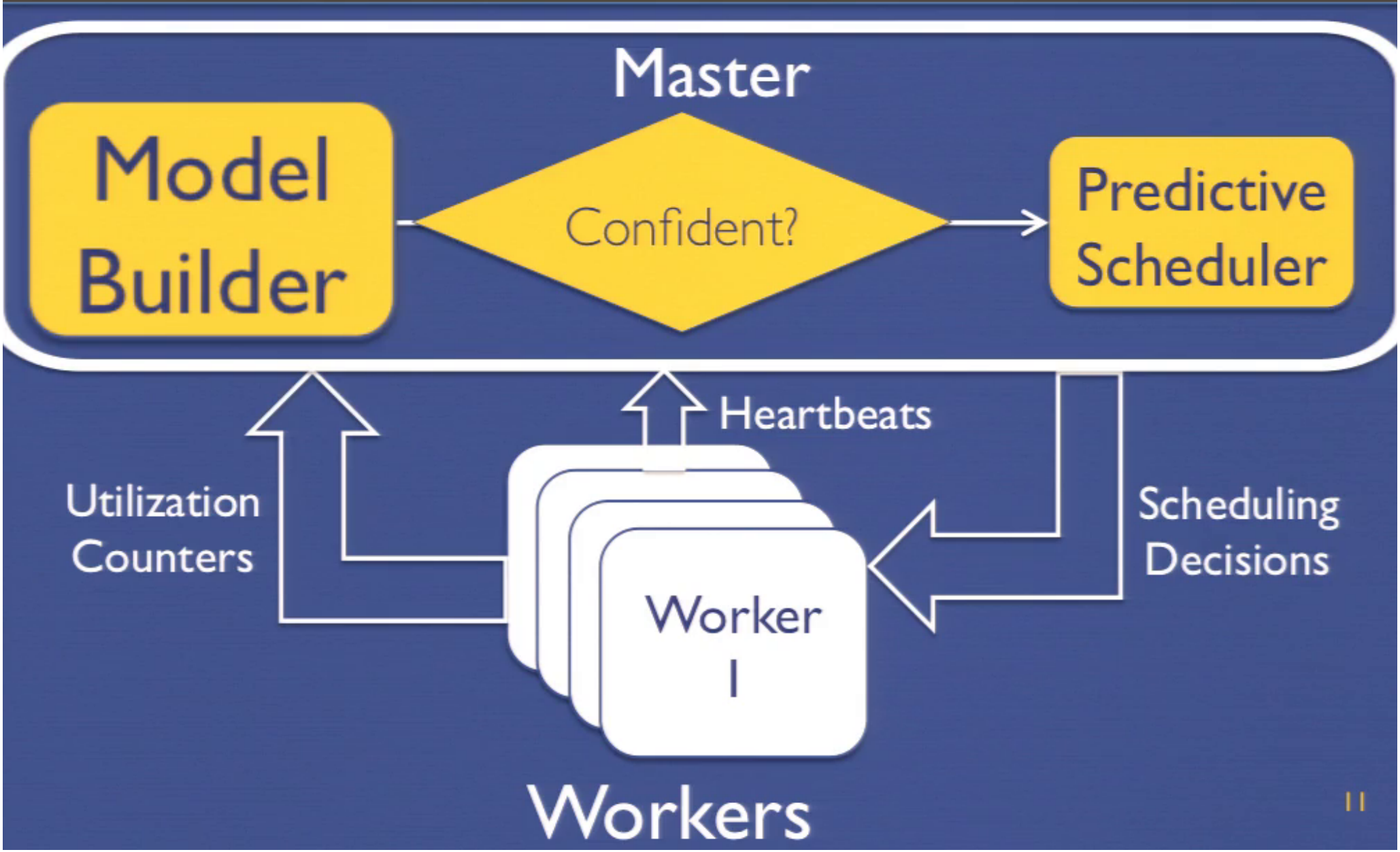
Existing *Proactive* Approach: Wrangler

Predict stragglers to avoid them



Existing *Proactive* Approach: Wrangler

Predict stragglers to avoid them



Wrangler: Classification for Predicting Stragglers

Build a model:

Predict:

Wrangler: Classification for Predicting Stragglers

Build a model:

{Utilization Counters, Straggler/Non-Straggler} → Learning → Classifier

A training data point corresponds to a task executed by a node

Wrangler: Classification for Predicting Stragglers

For every node!

Build a model:

{Utilization Counters, Straggler/Non-Straggler} → Learning → Classifier

A training data point corresponds to a task executed by a node

Wrangler: Classification for Predicting Stragglers

For every node!

For every workload!!

Build a model:

{Utilization Counters, Straggler/Non-Straggler} → Learning → Classifier

A training data point corresponds to a task executed by a node

Why model every {node, workload} pair?

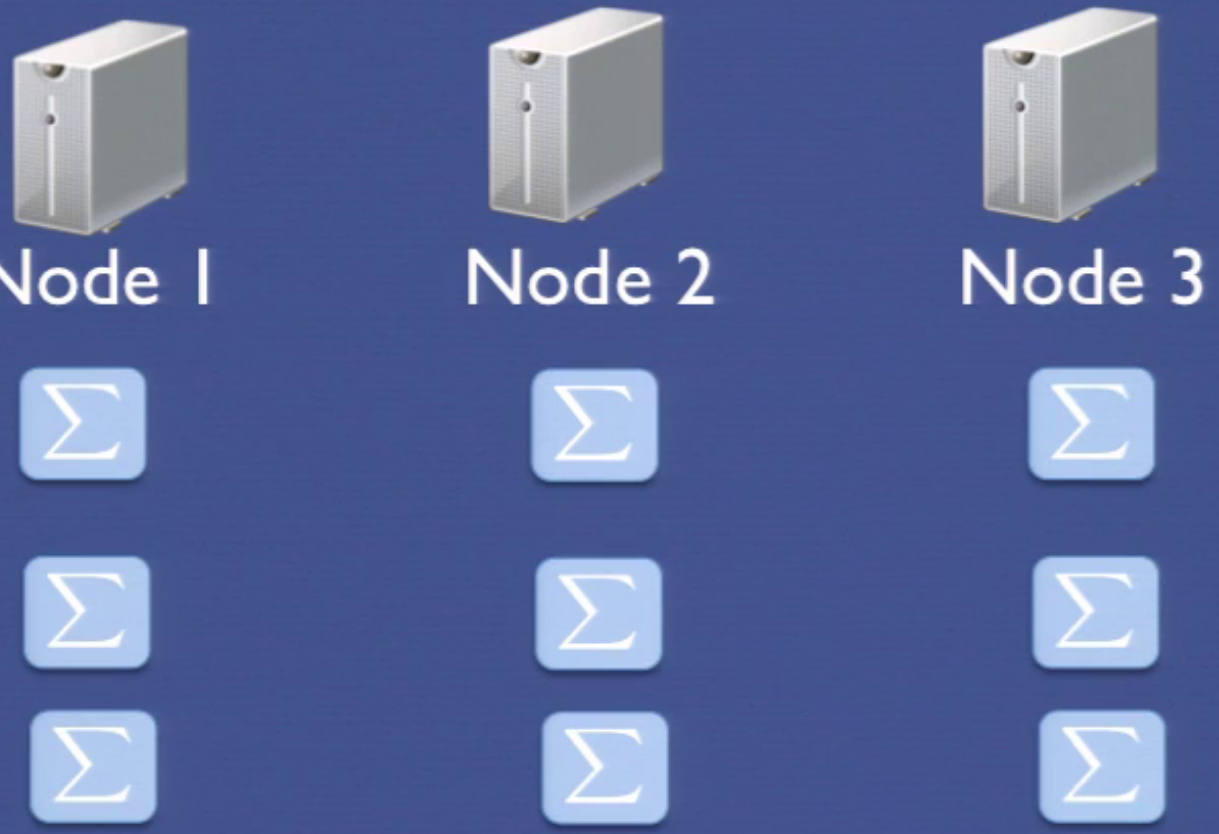
Key observation

Straggler causing factors vary across nodes and across time!

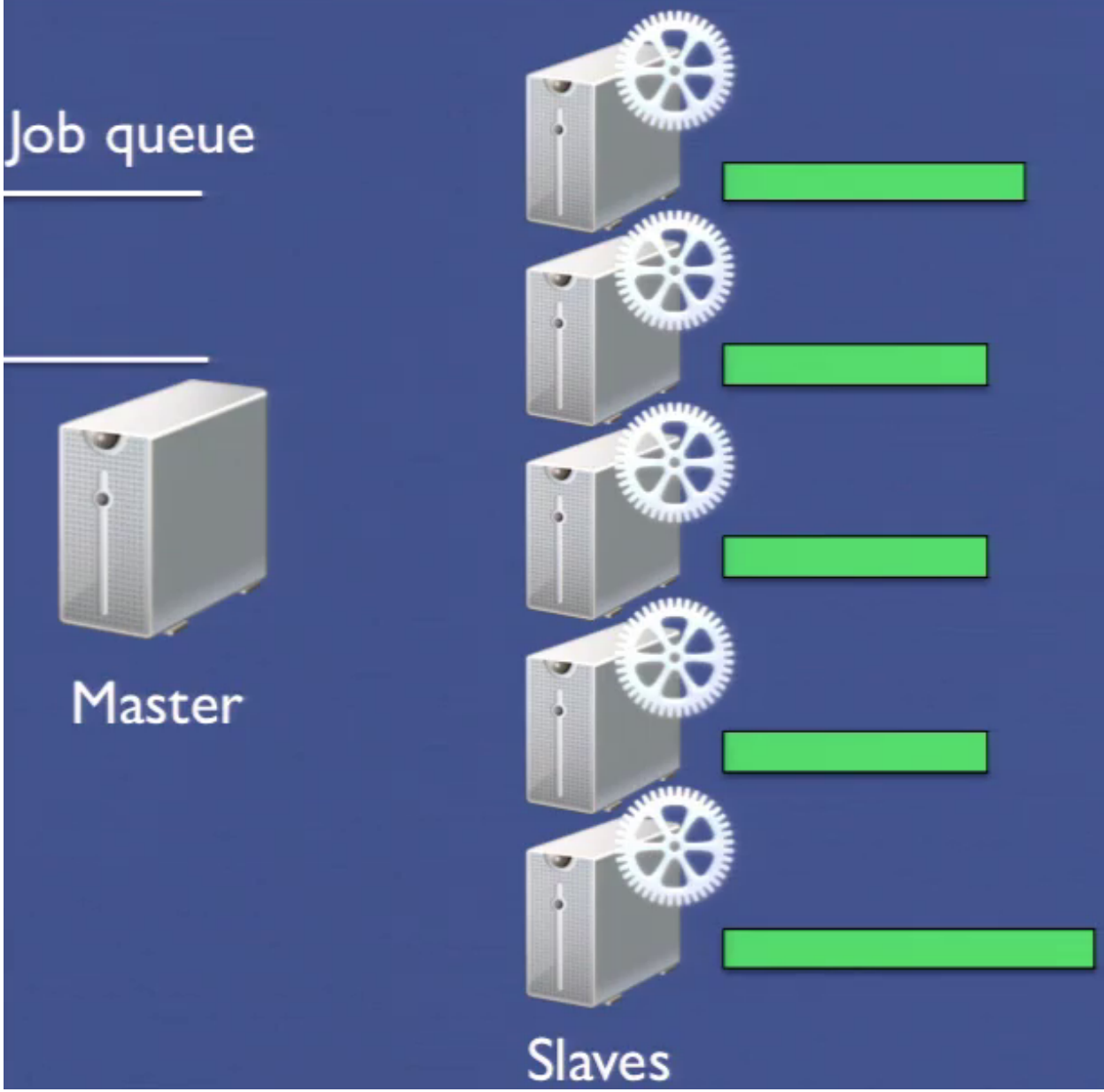
Why?

- Complex task-to-node interactions
- Complex task-to-task interactions
- Heterogeneous clusters
- Heterogeneous task requirements

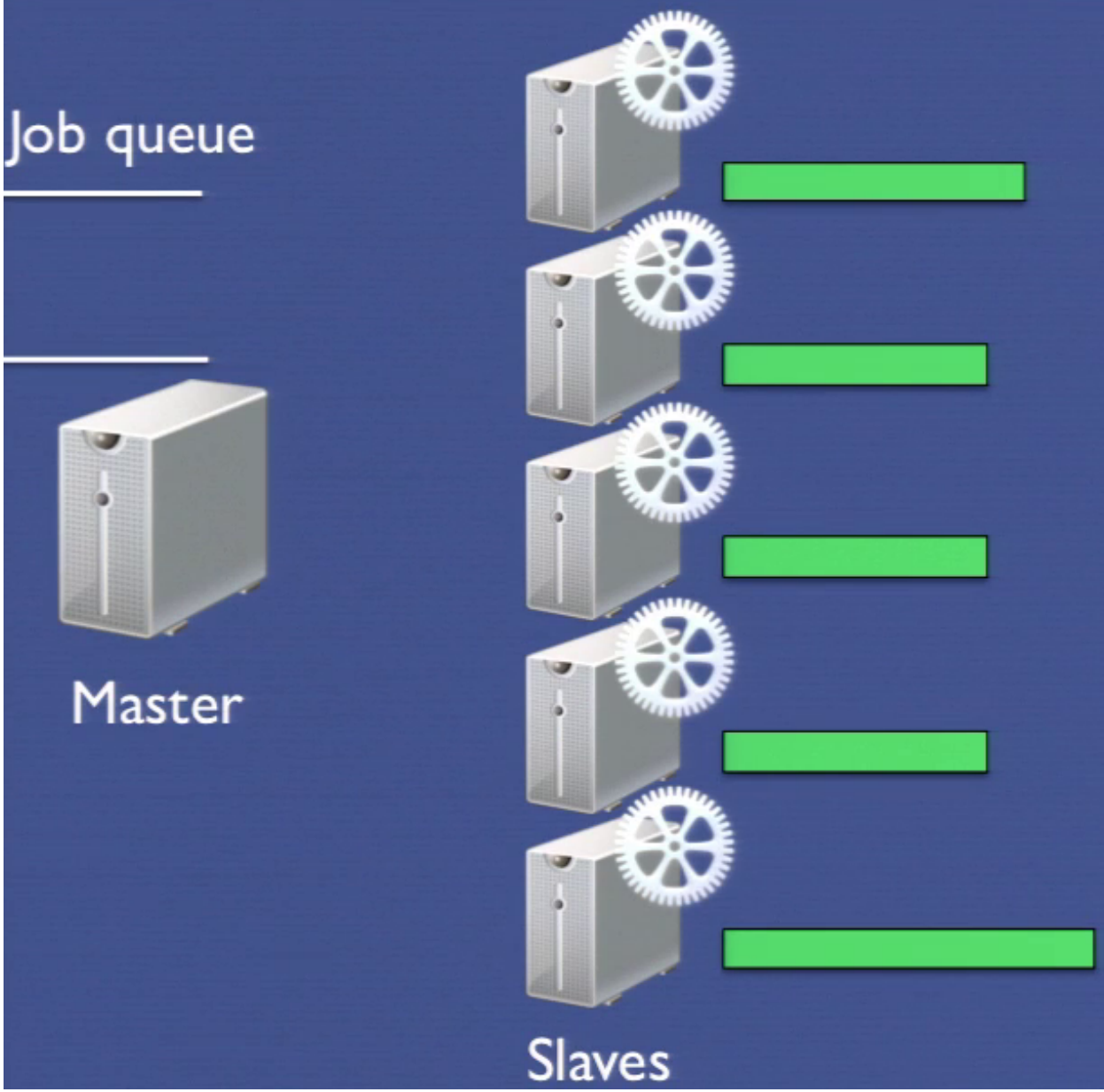
So, model every {node, workload} pair!



Model Builder: Training data collection



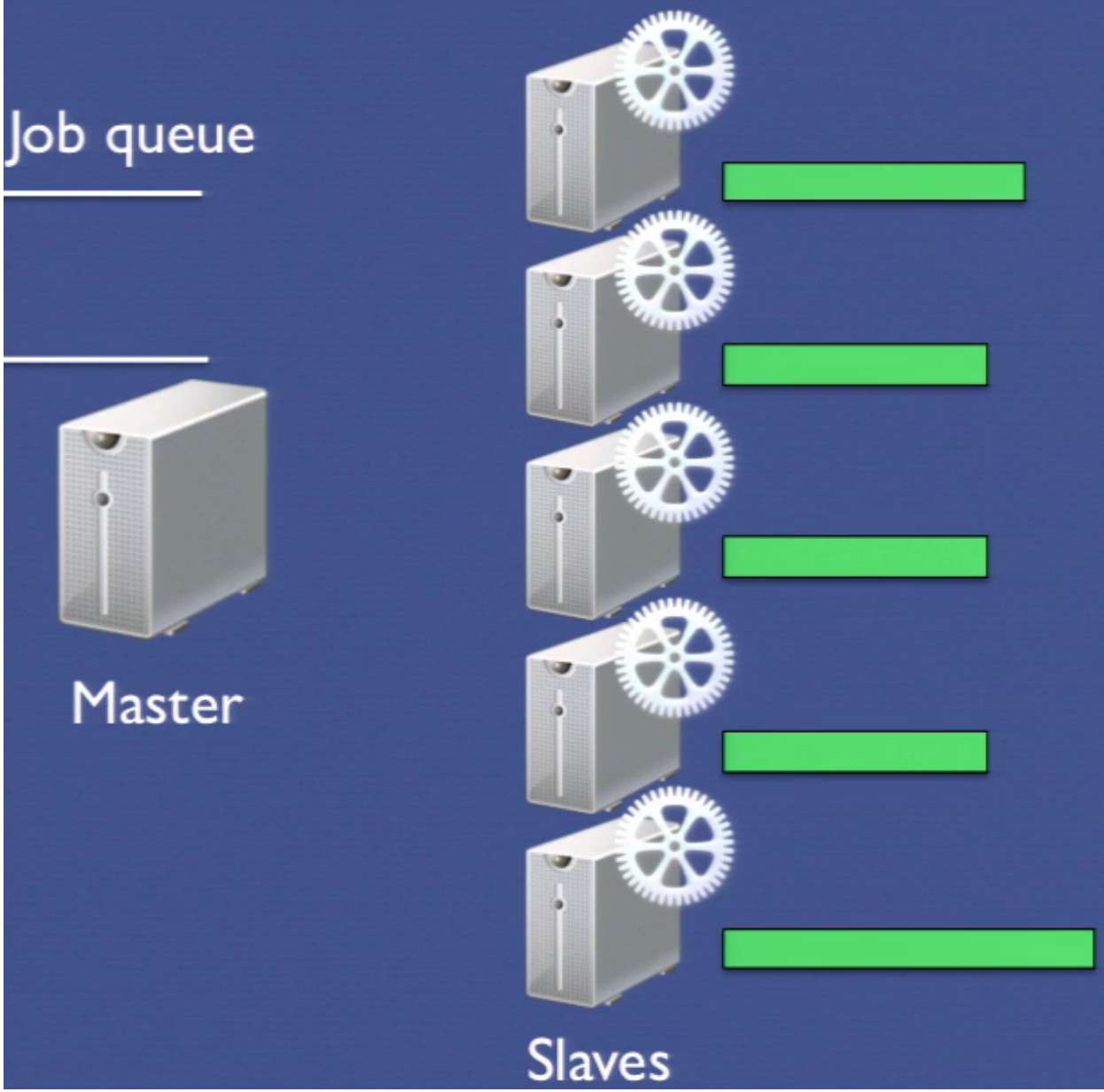
Model Builder: Training data collection



Workload A

5 models

Model Builder: Training data collection



Workload A

5 models

A *single* training data point per node

However….

Real-world production clusters could contain over 1000 nodes

- Scalability!
  - Need to train too many models separately
  - Prohibitively long training data capture duration

Outline

- ☑ Problem Context
- ☑ Existing Approaches:
  - ☑ Reactive: Speculative Execution
  - ☑ Proactive: Predictive modeling based approaches
- ❑ A New MTL Formulation
- ❑ Application to Straggler Avoidance
- ❑ Evaluation

Our Proposal

Observations:

- Underlying modeling task remains the same
- Learning from other similar tasks should help
  - Reduce training data capture time
  - Improve accuracy by generalizing better

Our Proposal

Observations:

- Underlying modeling task remains the same
- Learning from other similar tasks should help
  - Reduce training data capture time
  - Improve accuracy by generalizing better

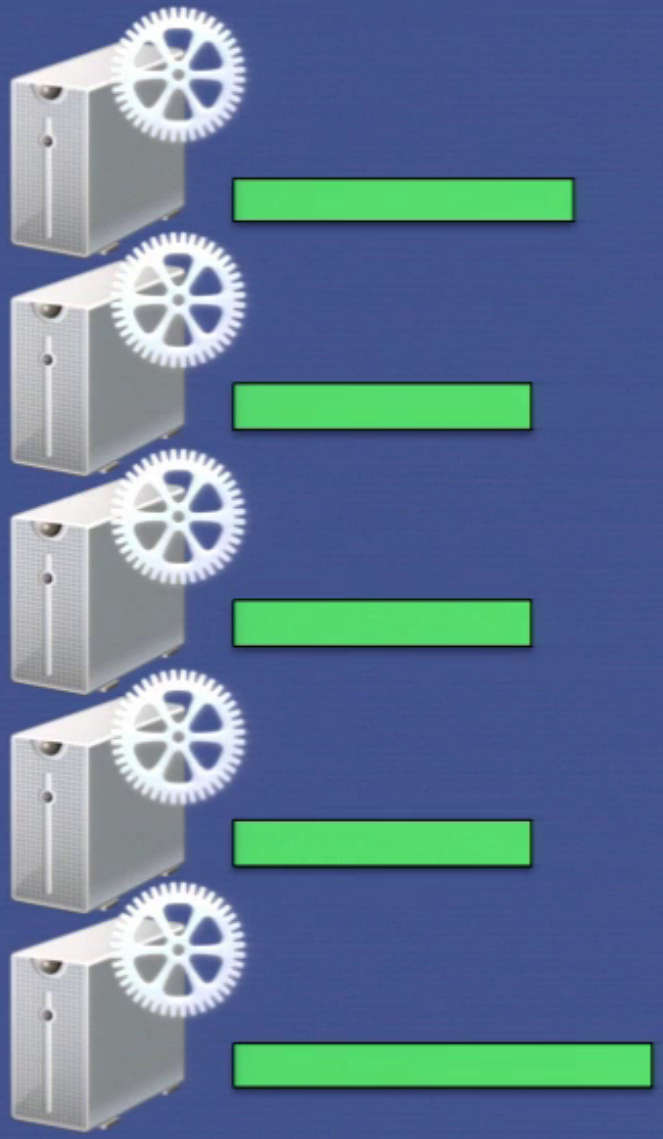
Idea

Share data across nodes and workloads:
Multi Task Learning

Model Builder: Training data collection

Job queue

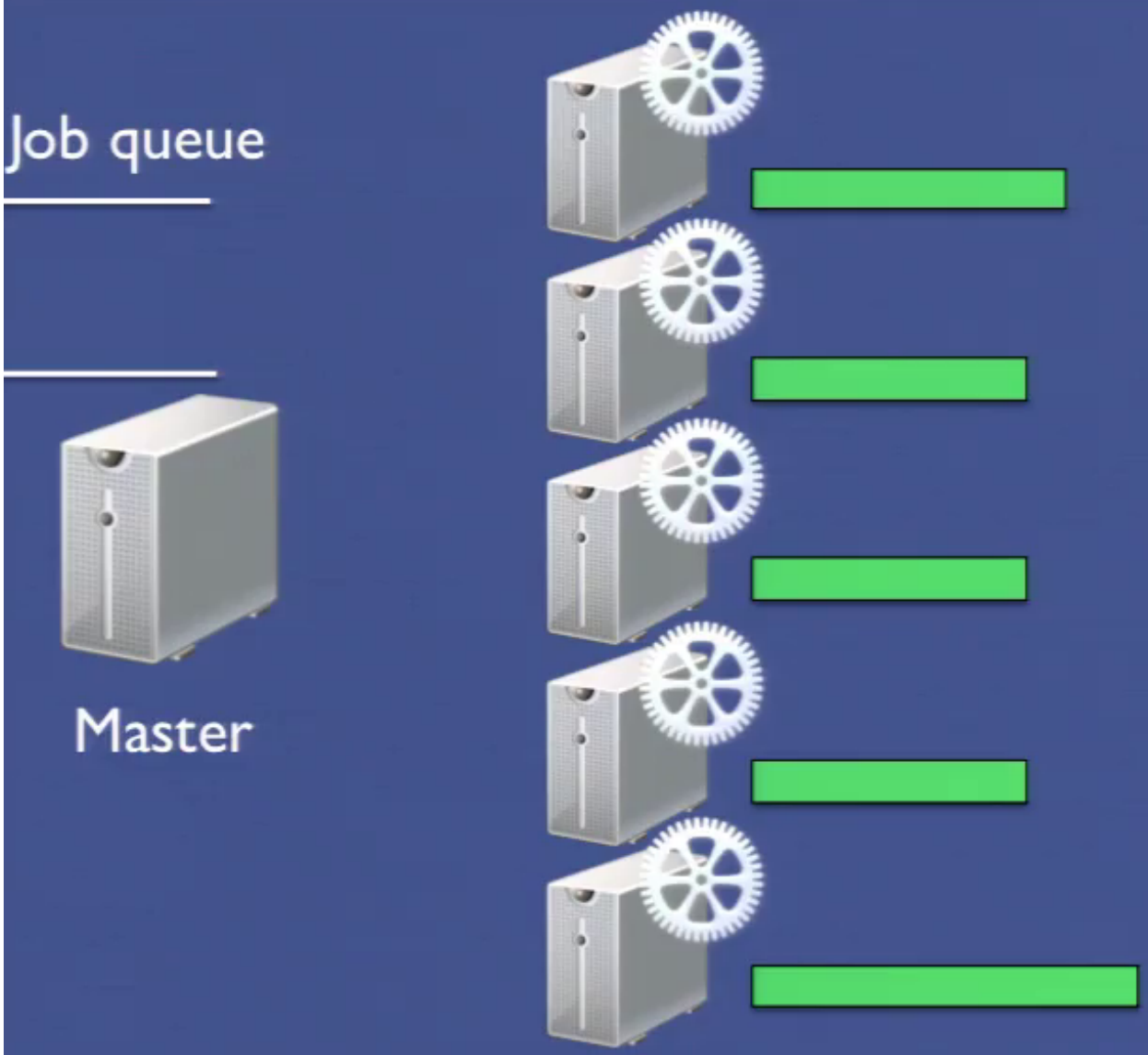
Master



Workload A

A *single* training data point per node

Model Builder: Training data collection



Regularized Multi-Task Learning*

- T learning tasks
- Instead of one w, we need to learn a w for each of the T tasks

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$$

Common across all the learning tasks

Specific for a learning tasks, *t*

*Evgeniou, et al., KDD 2004

Regularized Multi-Task Learning*

- T learning tasks
- Instead of one w, we need to learn a w for each of the T tasks

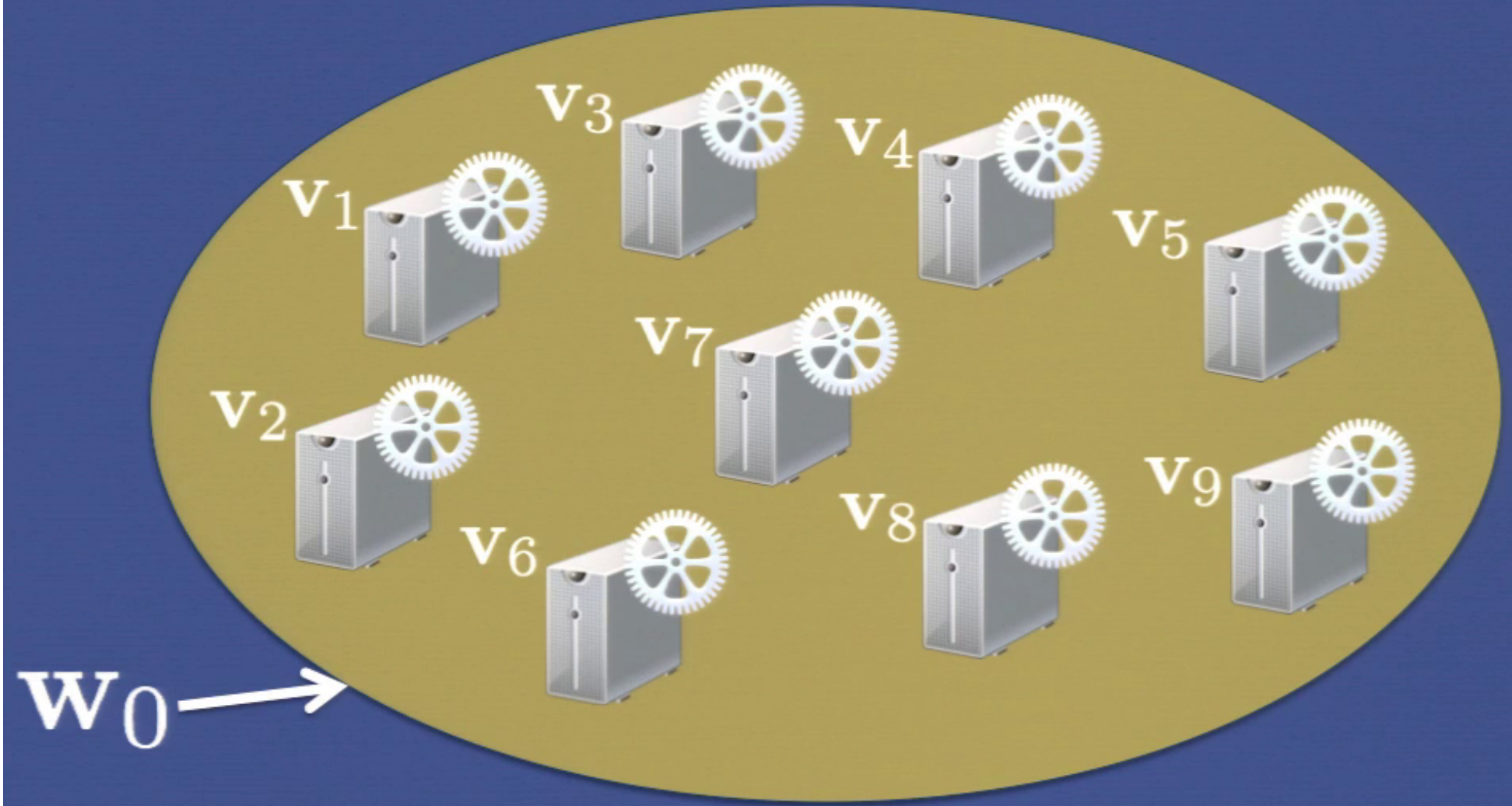
$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$$

$$\min_{\mathbf{w}_0, \mathbf{v}_t, b} \lambda_0 \|\mathbf{w}_0\|^2 + \frac{\lambda_1}{T} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2 + \text{Loss function}$$

*Evgeniou, et al., KDD 2004

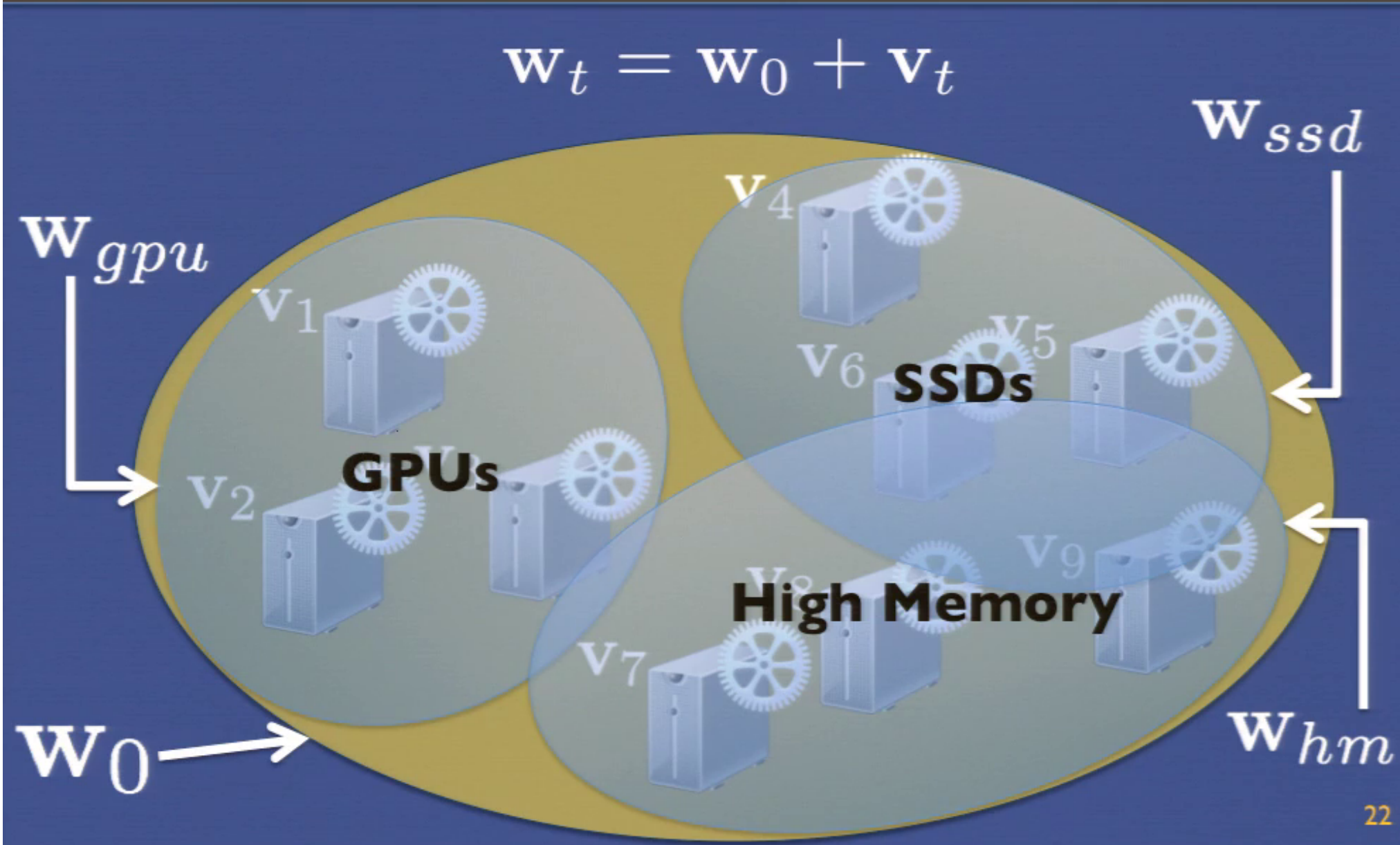
Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$$



Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$$



Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$$

Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_g$$

Common across the tasks in a group, denoted by g

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_{gpu} + \mathbf{w}_{ssd} + \ldots.$$

Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_g$$

Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_g$$

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \sum_{p=1}^{P} \underbrace{\mathbf{w}_{p,g_p(t)}}$$

Weight vector of the g-th group of the p-th partition

Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_g$$

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \sum_{p=1}^{P} \underbrace{\mathbf{w}_{p,g_p(t)}}$$

All tasks belong to the same group

Weight vector of the g-th group of the p-th partition

Proposed Formulation

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \mathbf{w}_g$$

$$\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t + \sum_{p=1}^{P} \mathbf{w}_{p,g_p(t)}$$

All tasks belong to the same group

Each task is its own group

Weight vector of the g-th group of the p-th partition

$$\mathbf{w}_t = \sum_{p=1}^{P} \mathbf{w}_{p,g_p(t)}$$

Proposed Formulation

$$\min_{\mathbf{w}_{p,g},b} \sum_{p=1}^{P} \sum_{g=1}^{G_p} \lambda_{p,g} ||\mathbf{w}_{p,g}||^2 + \text{Loss function}$$

Reduction to a Standard SVM

With an appropriate change of variable,

$$\min_{\tilde{\mathbf{w}},b} \lambda\|\tilde{\mathbf{w}}\|^2 + \sum_{t=1}^{T}\sum_{i=1}^{m_t} \xi_{it}$$

s.t.

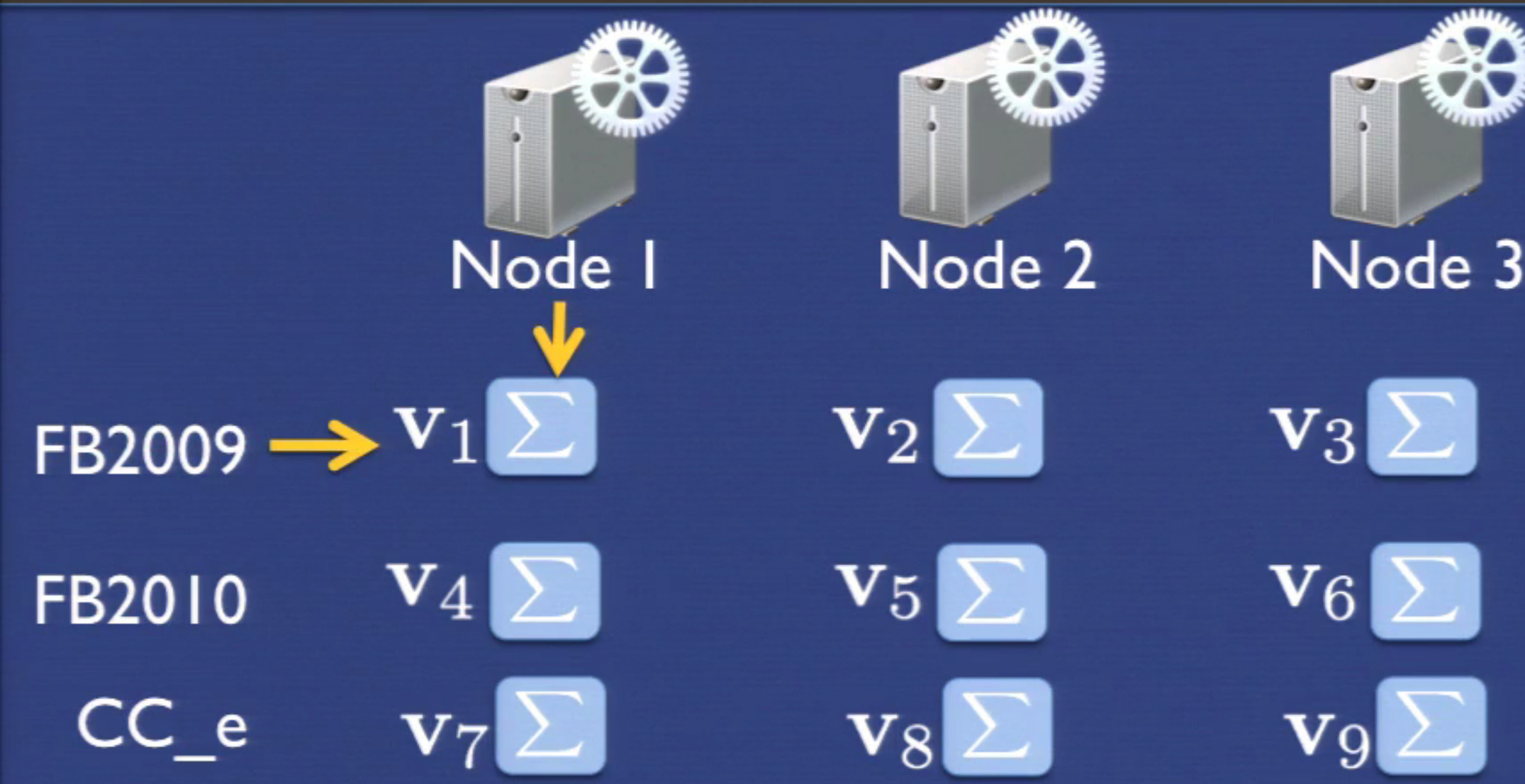
$$y_{it}(\tilde{\mathbf{w}}^T \phi(\mathbf{x}_{it}) + b) \geq 1 - \xi_{it} \quad \forall i, t$$

$$\xi_{it} \geq 0 \quad \forall i, t$$

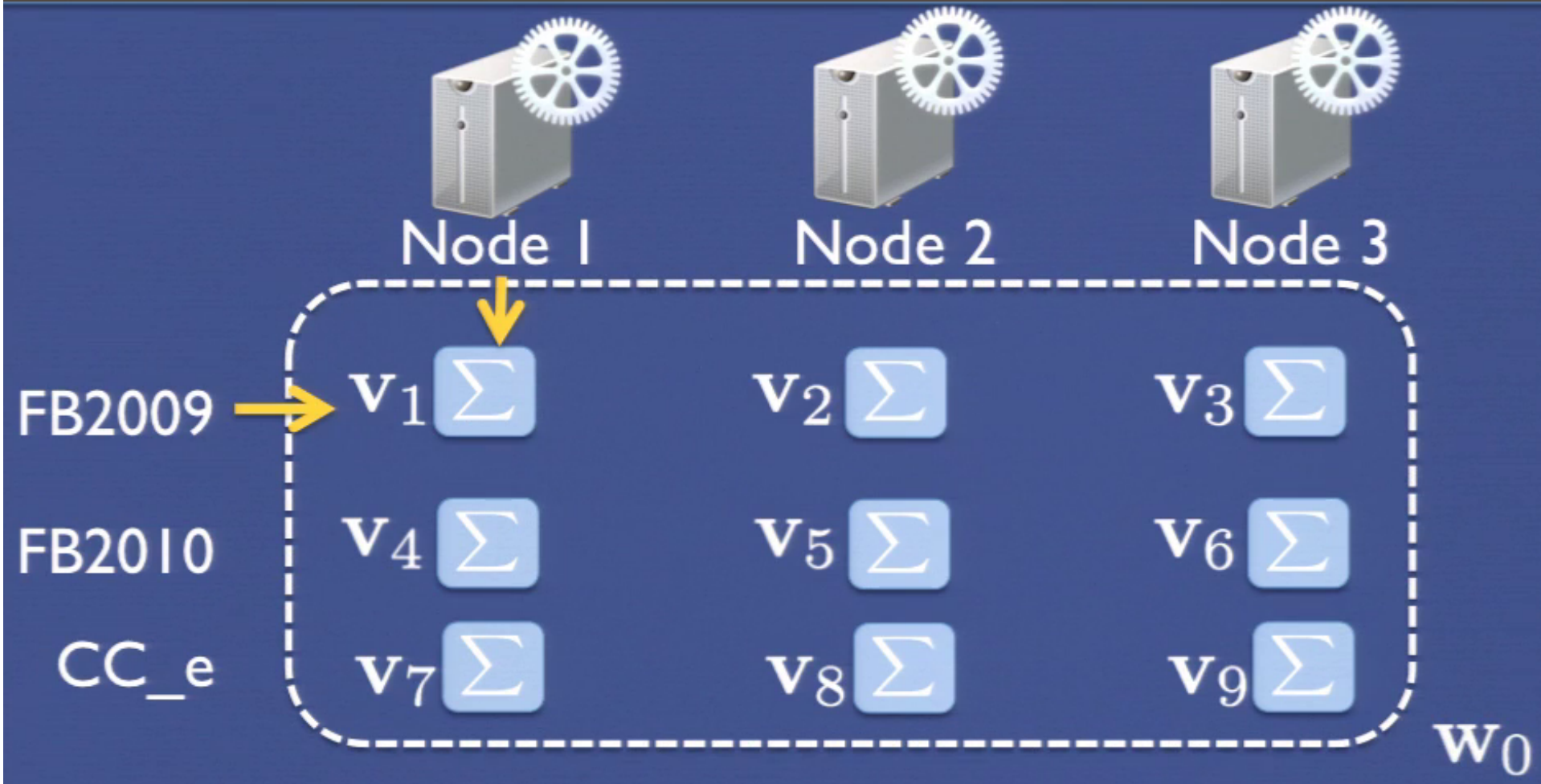
Outline

- ☑ Problem Context
- ☑ Existing Approaches:
  - ☑ Reactive: Speculative Execution
  - ☑ Proactive: Predictive modeling based approaches
- ☑ A New MTL Formulation
- ❑ Application to Straggler Avoidance
- ❑ Evaluation

Application to straggler avoidance:

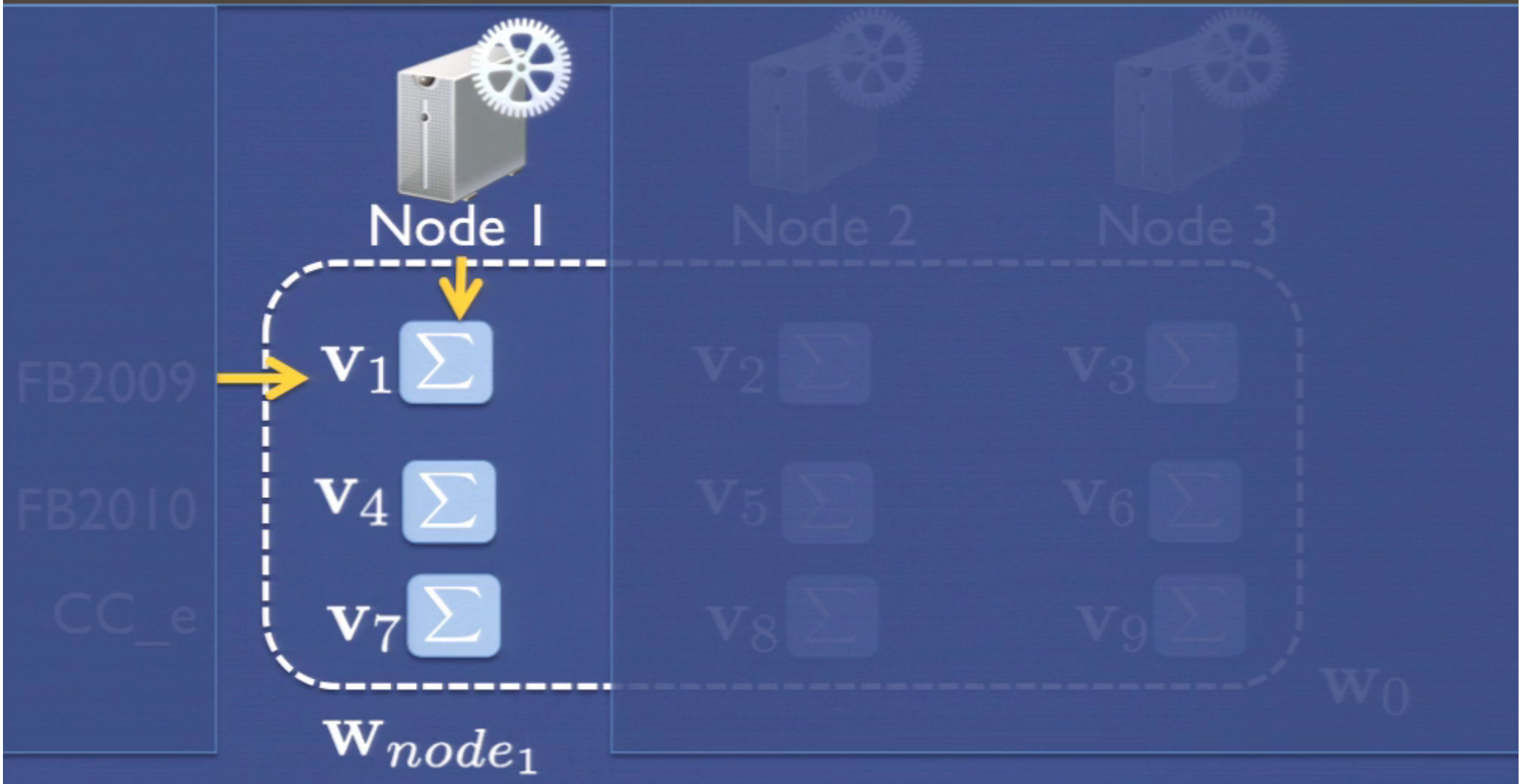


Application to straggler avoidance:



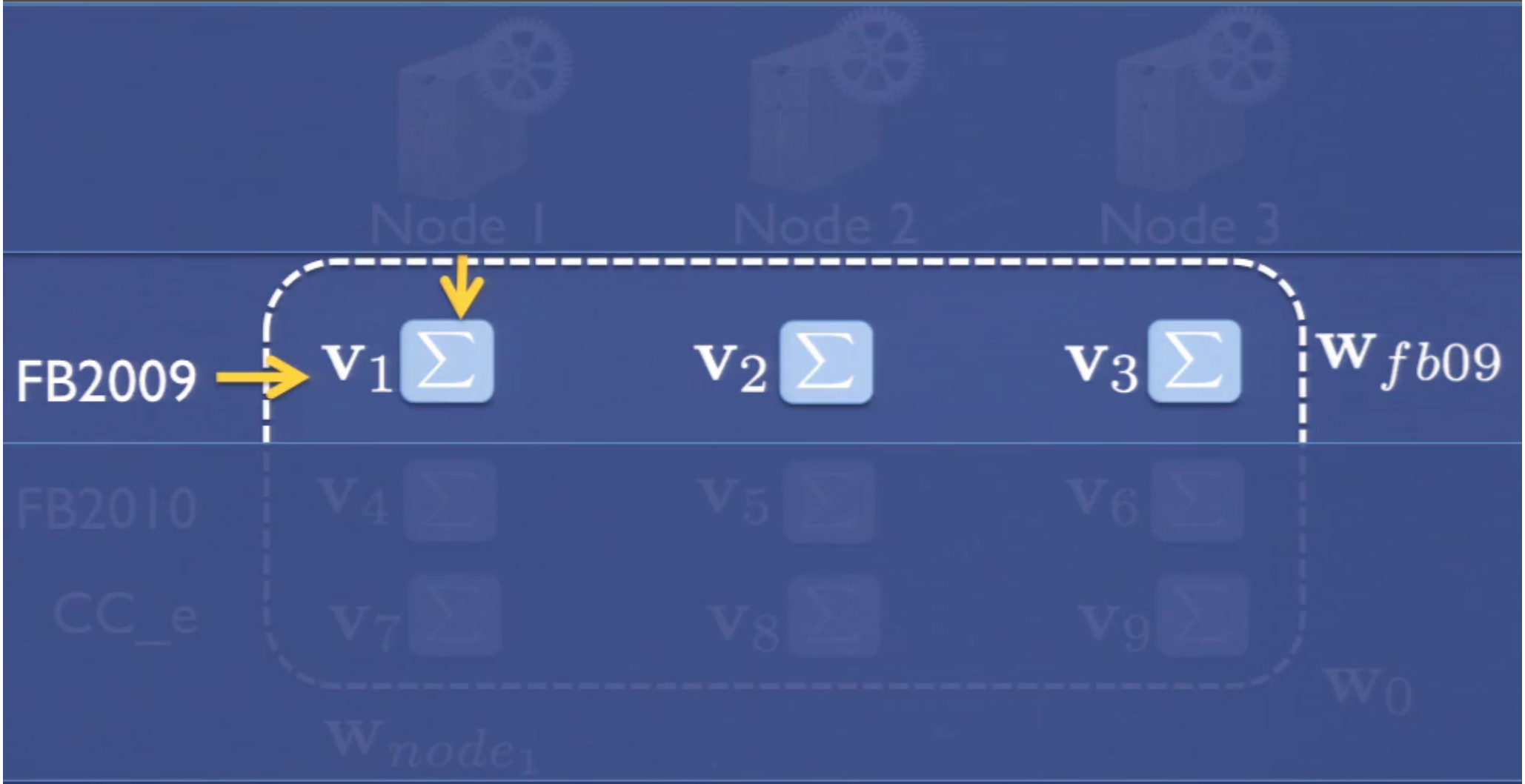
$$\mathbf{w}_1 = \mathbf{w}_0 +$$

Application to straggler avoidance:



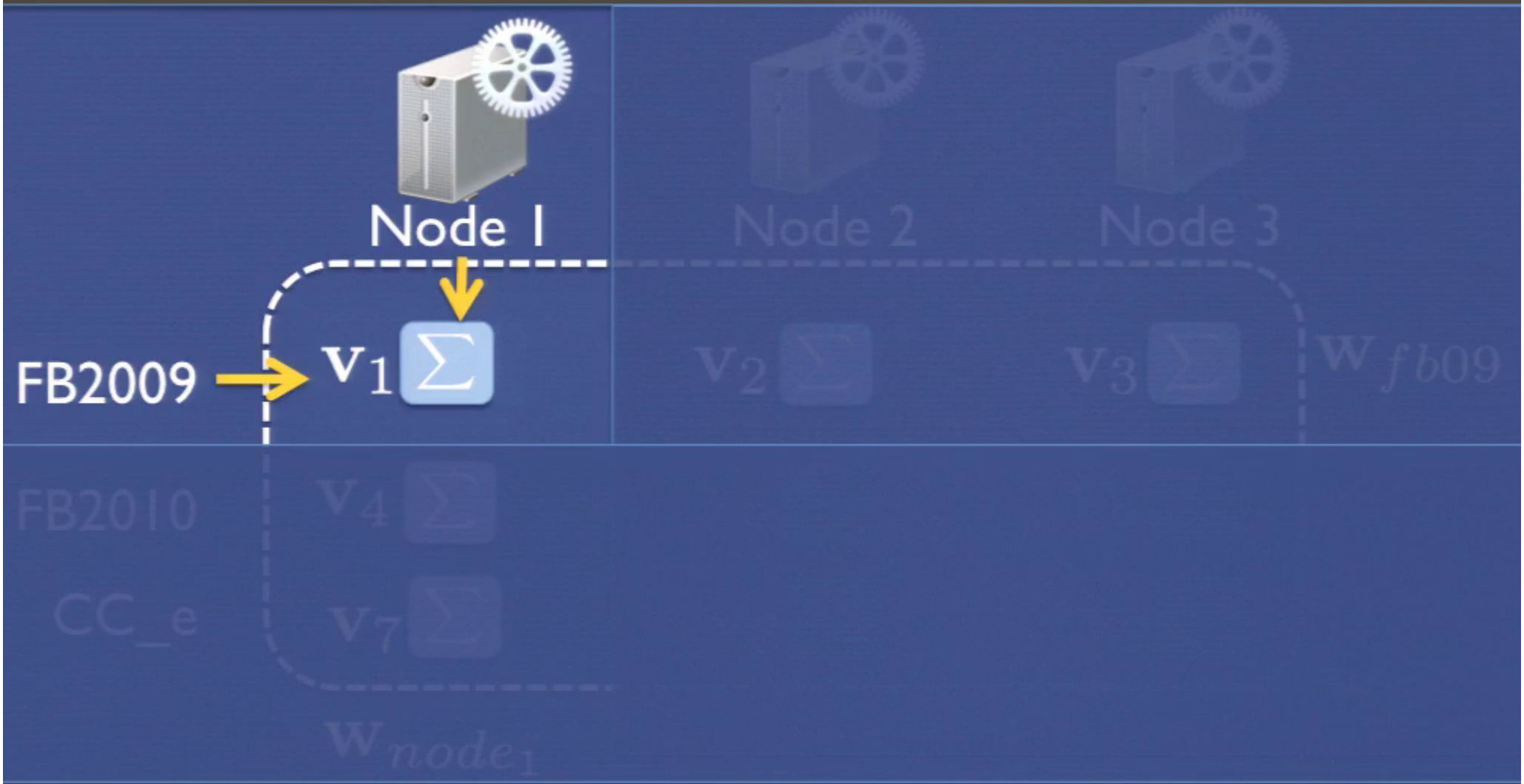
$$\mathbf{w}_1 = \mathbf{w}_0 + \mathbf{w}_{node_1} +$$

Application to straggler avoidance:



$$\mathbf{w}_1 = \mathbf{w}_0 + \mathbf{w}_{node_1} + \mathbf{w}_{fb09} +$$

Application to straggler avoidance:



$$\mathbf{w}_1 = \mathbf{w}_0 + \mathbf{w}_{node_1} + \mathbf{w}_{fb09} + \mathbf{v}_1$$

Proposed Formulation: Predicting Stragglers

The corresponding training problem is then,

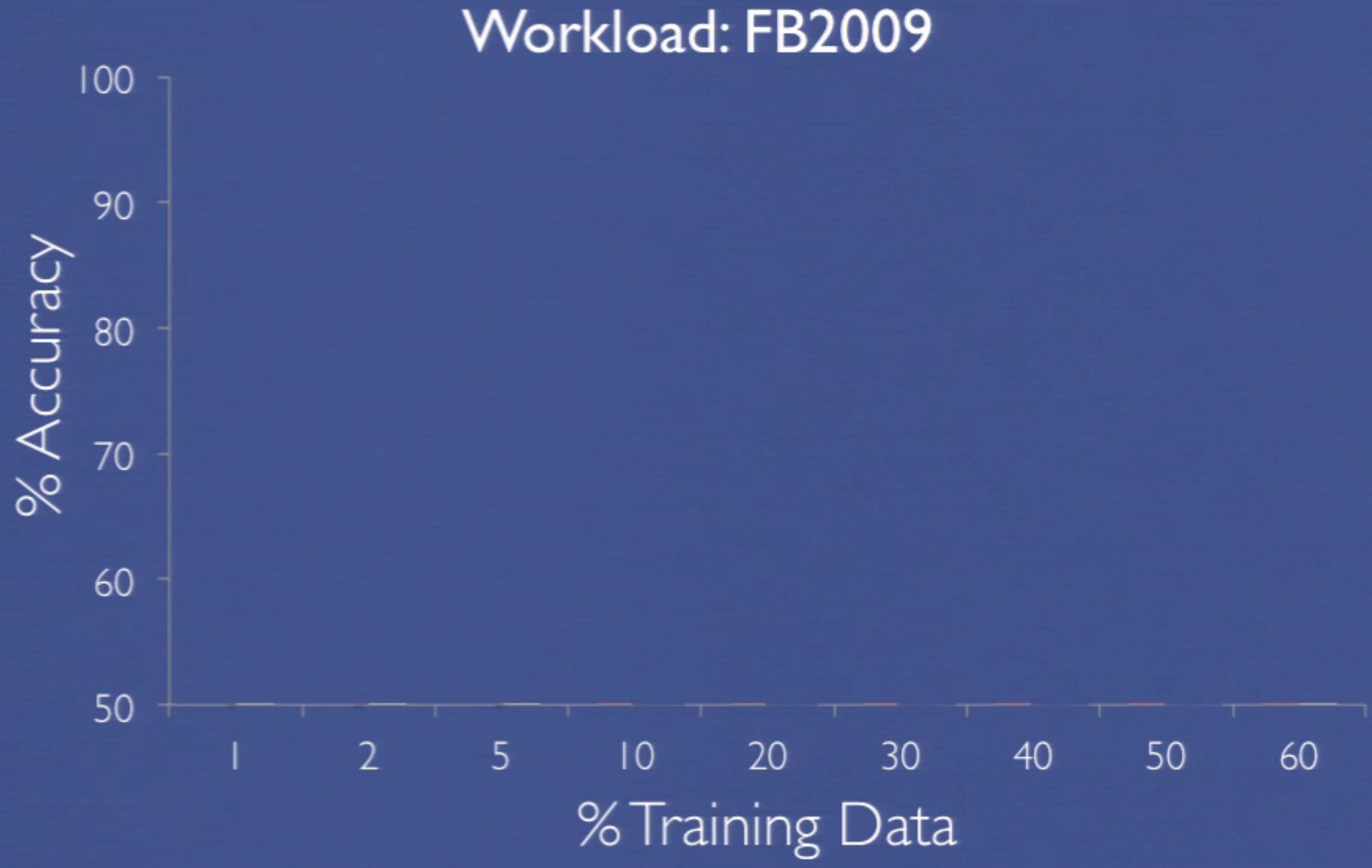
$$\min_{\mathbf{w},b} \lambda_0 \|\mathbf{w}_0\|^2 + \frac{\nu}{N}\sum_{n=1}^{N}\|\mathbf{w}_n\|^2 + \frac{\omega}{L}\sum_{l=1}^{L}\|\mathbf{w}_l\|^2$$

$$+ \frac{\tau}{T}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + \text{Loss function}$$

Outline

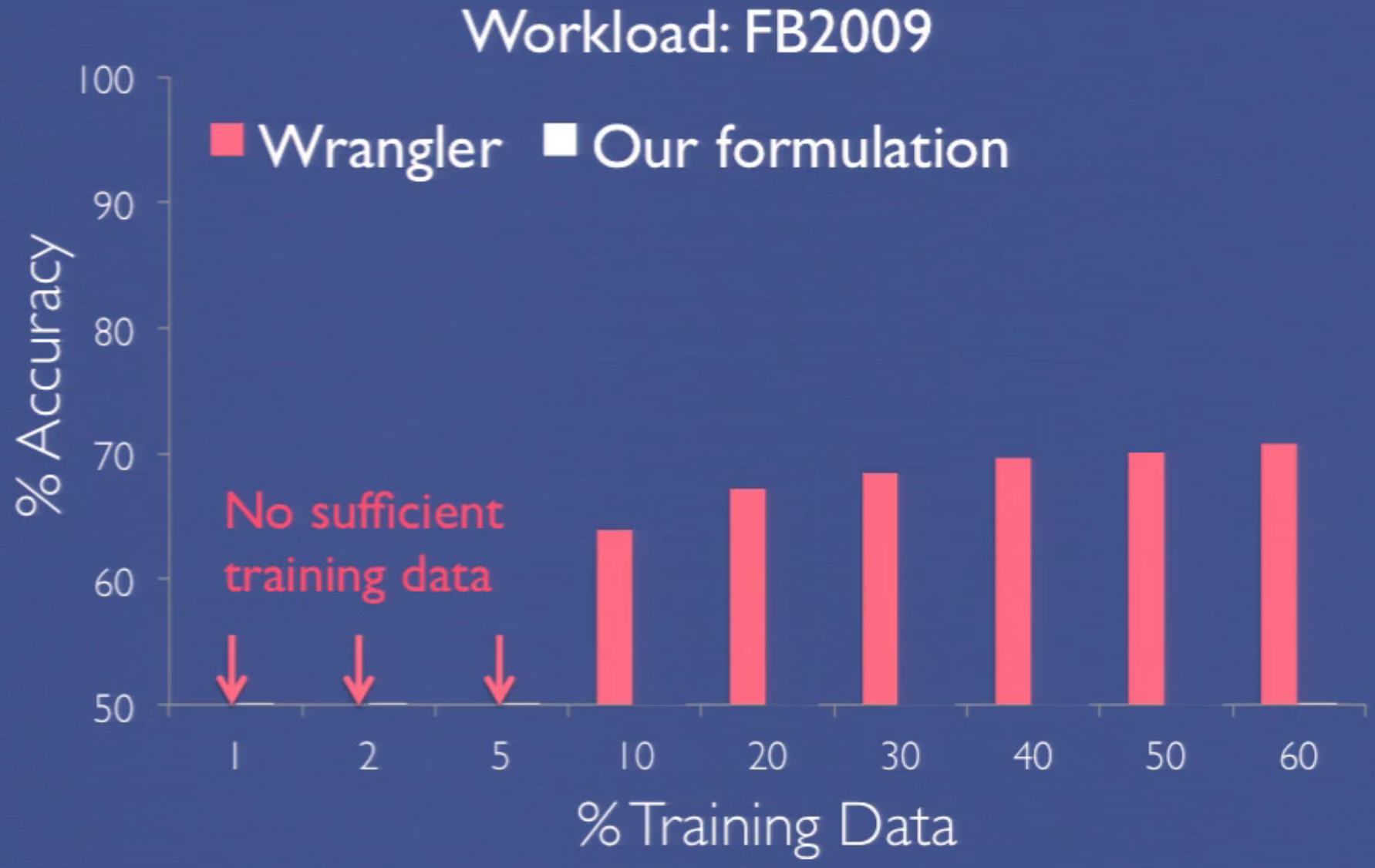
- ☑ Problem Context
- ☑ Existing Approaches:
  - ☑ Reactive: Speculative Execution
  - ☑ Proactive: Predictive modeling based approaches
- ☑ A New MTL Formulation
- ☐ Application to Straggler Avoidance
- ☐ Evaluation

Evaluation 1: Prediction Accuracy



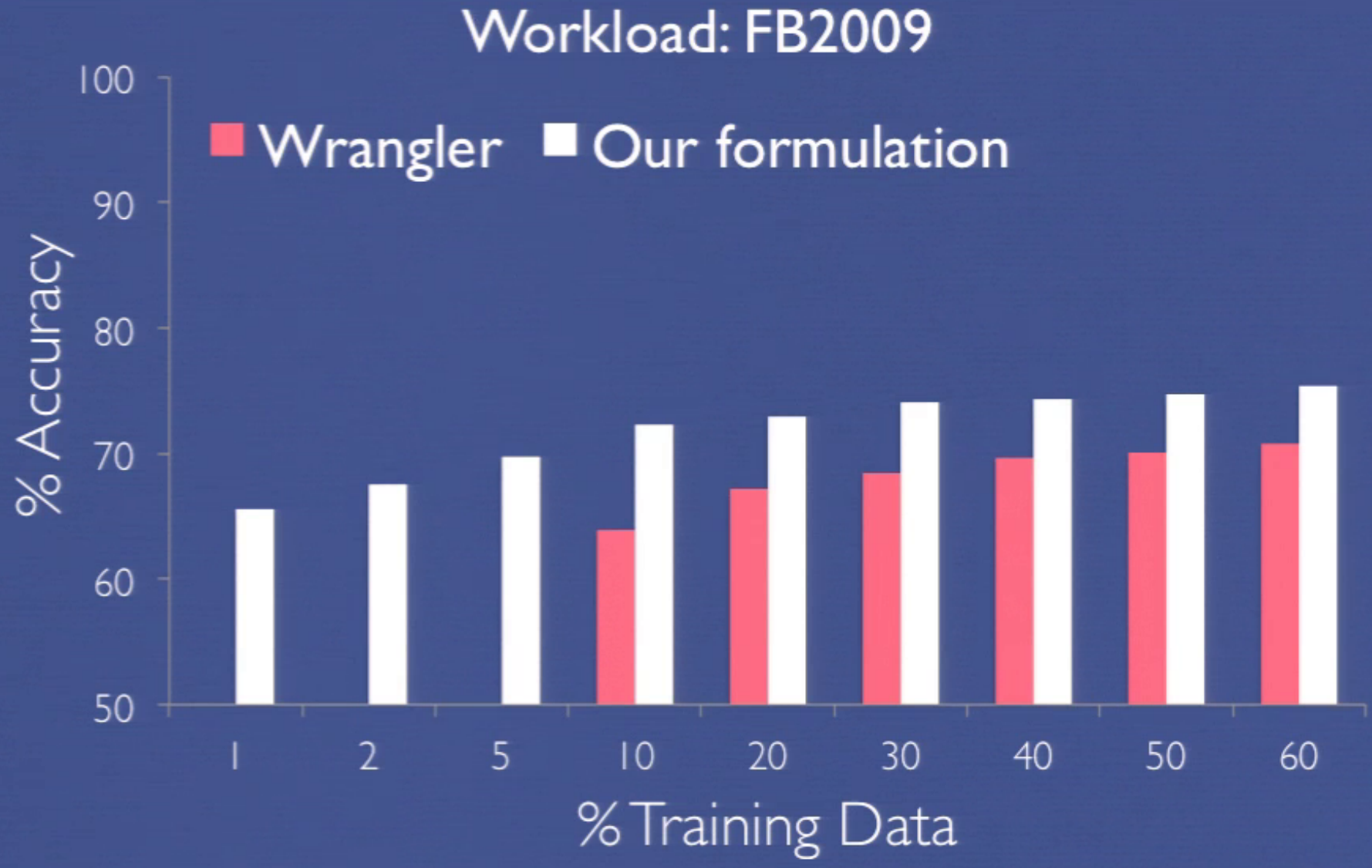
Note: Our dataset is balanced.

Evaluation 1: Prediction Accuracy



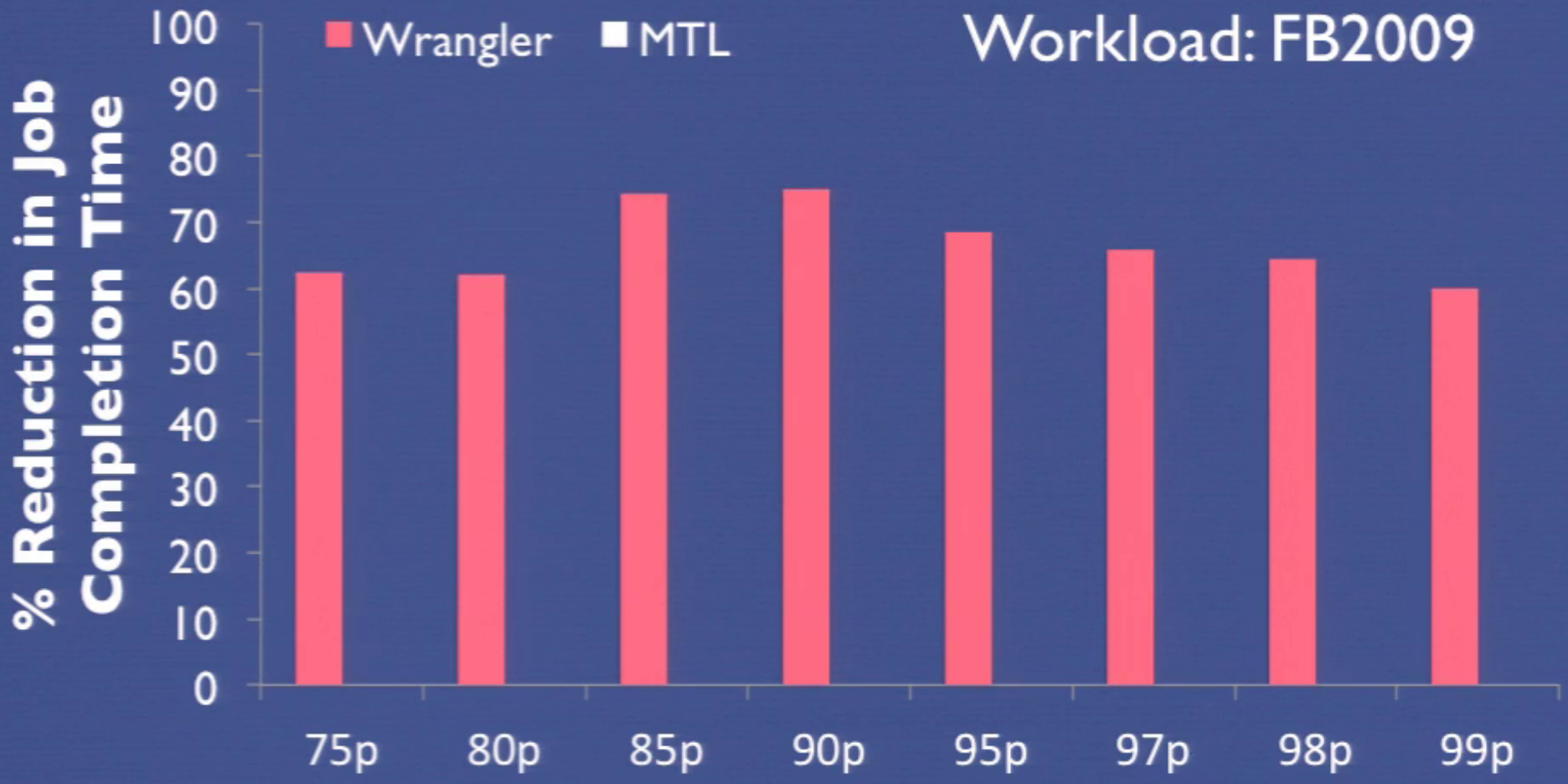
Note: Our dataset is balanced.

Evaluation 1: Prediction Accuracy

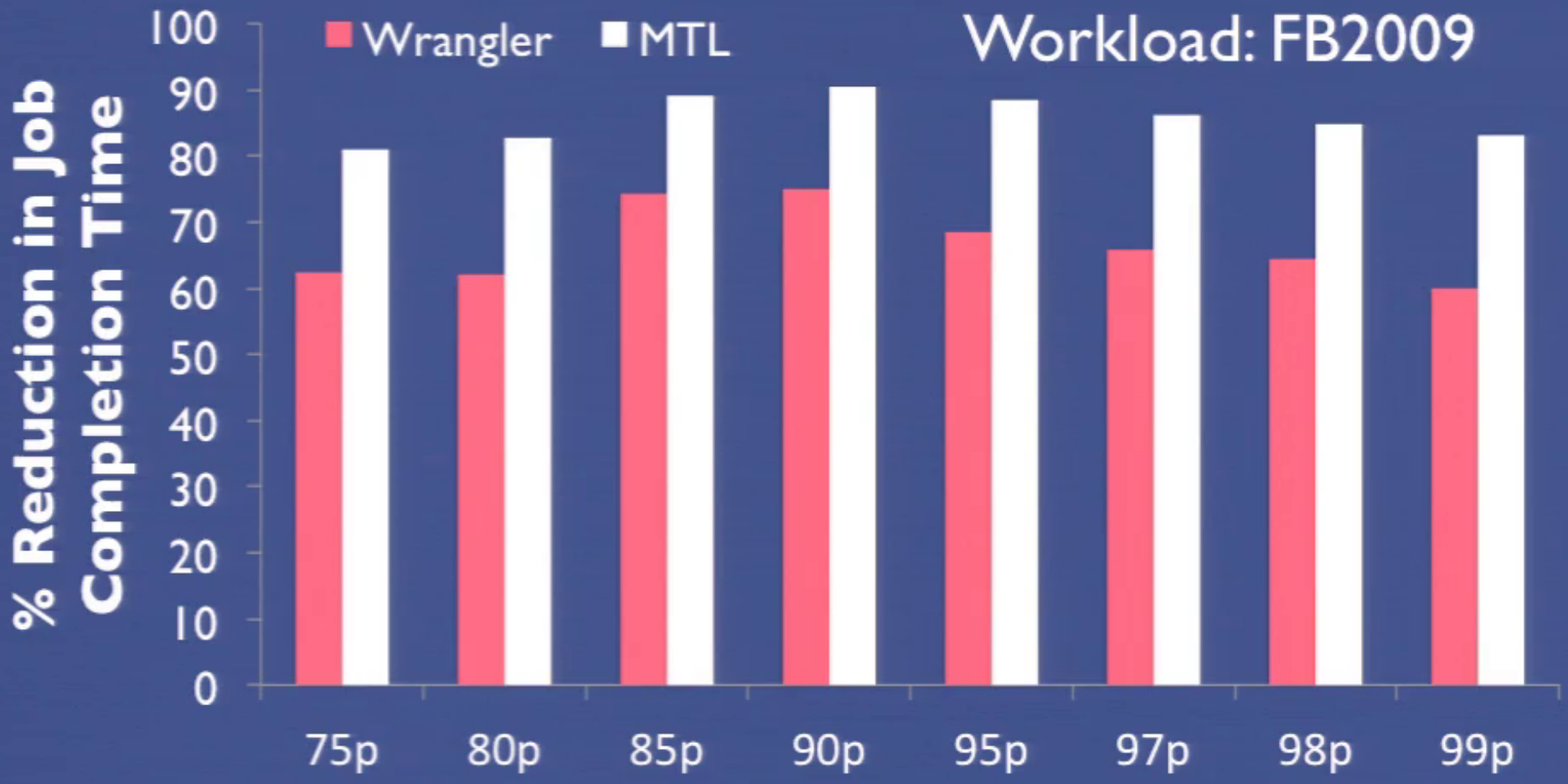


Note: Our dataset is balanced.

Evaluation II: Faster Jobs



Evaluation II: Faster Jobs



Evaluation II: Faster Jobs

We need only a sixth of training data!

Evaluation II: Faster Jobs

We need only a sixth of training data!

….i.e., 4 hours → 40 minutes!!

Conclusions

Proposed an MTL formulation that:

- Captures structure of learning tasks

Conclusions

Proposed an MTL formulation that:

- Captures structure of learning tasks

Showed Benefits of MTL on a real-world problem:

- Reduces job completion times further
- Generalizes better
- Needs only a sixth of training data