Concurrency Control for Scalable Bayesian Inference

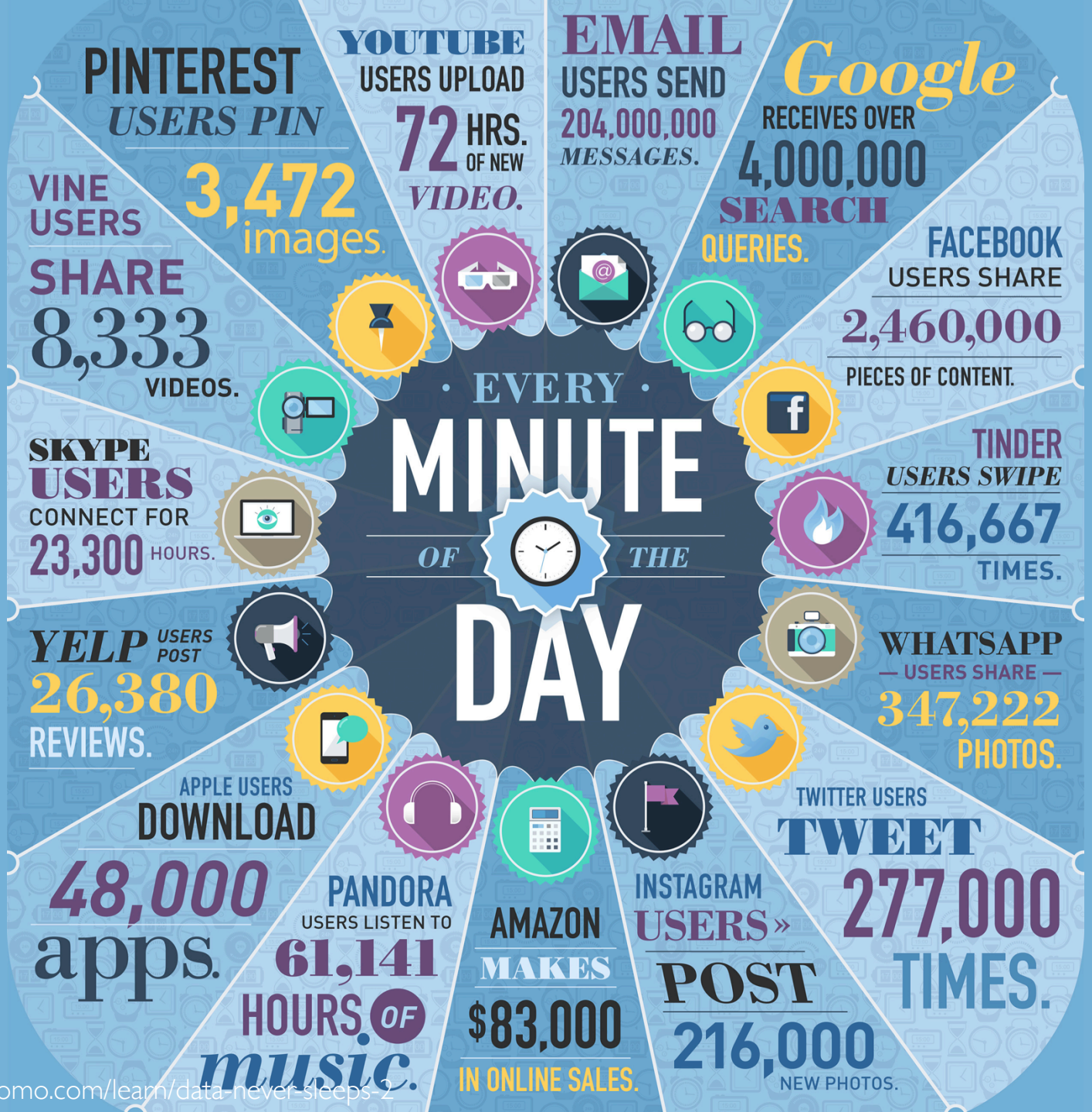
Joseph E. Gonzalez
Postdoc, UC Berkeley AMPLab
Co-founder, GraphLab Inc.
jegonzal@eecs.berkeley.edu

ISBA' 2014

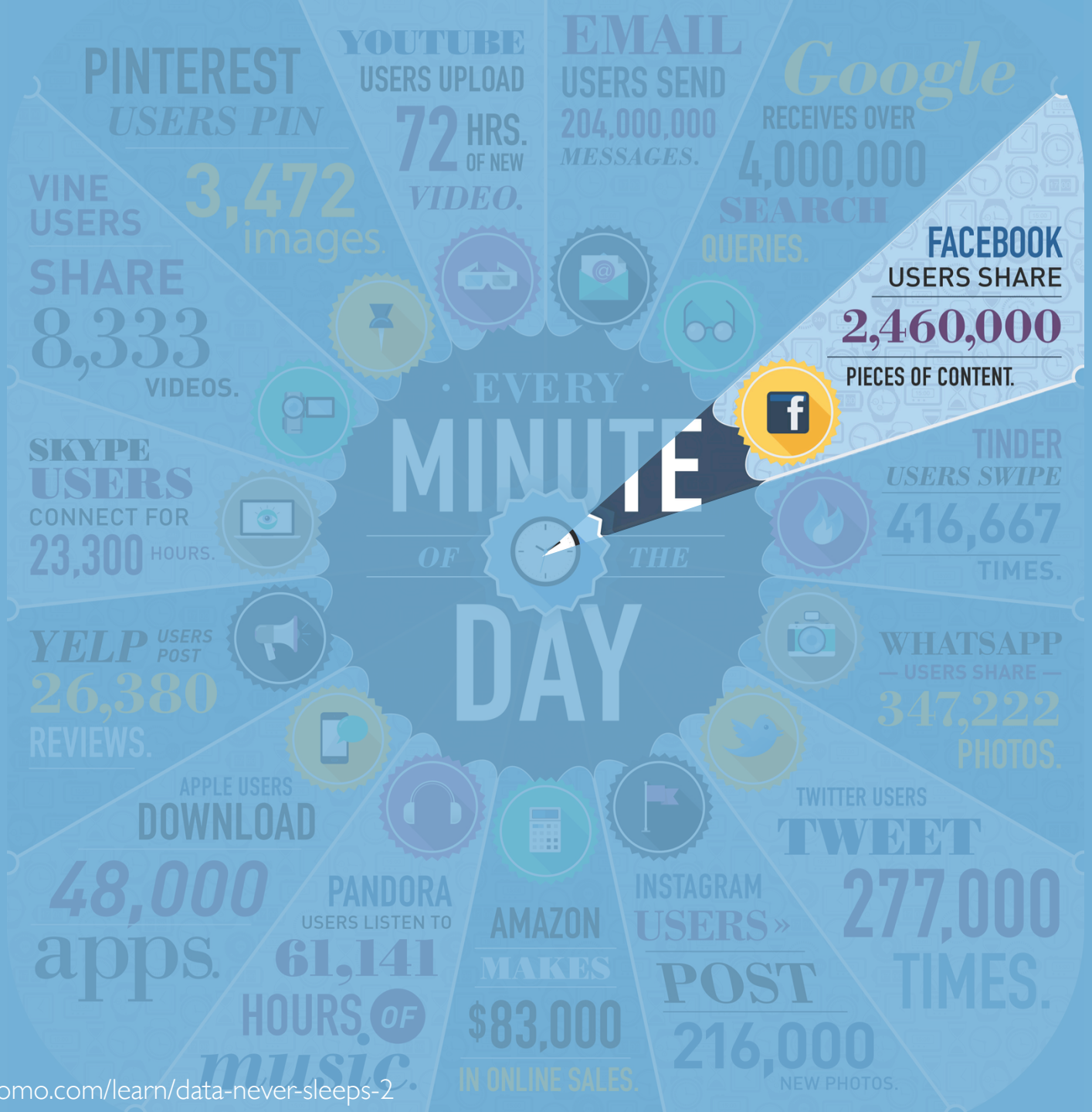
A Systems Approach to Scalable Bayesian Inference

Joseph E. Gonzalez
Postdoc, UC Berkeley AMPLab
Co-founder, GraphLab Inc.
jegonzal@eecs.berkeley.edu

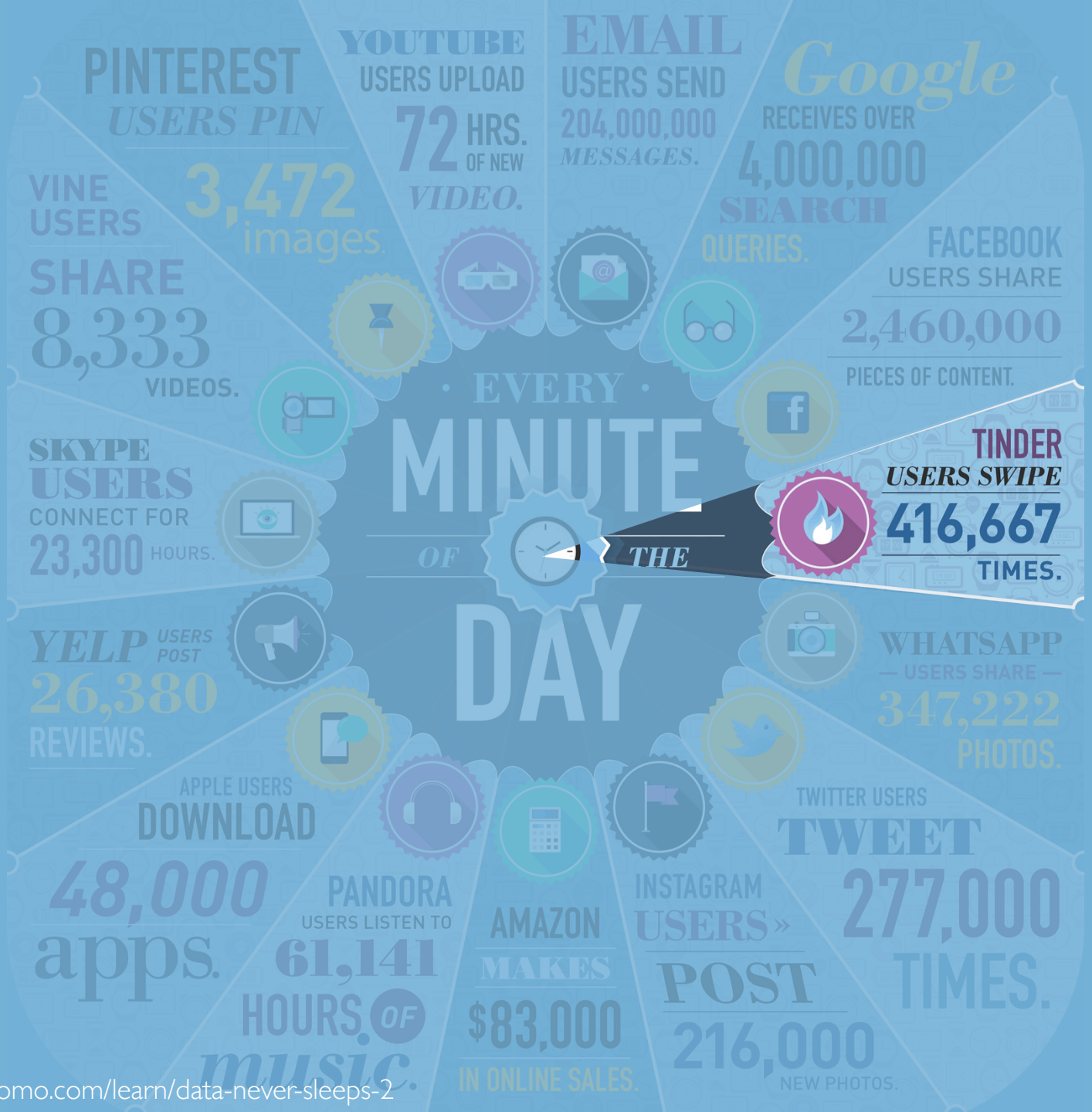
ISBA' 2014



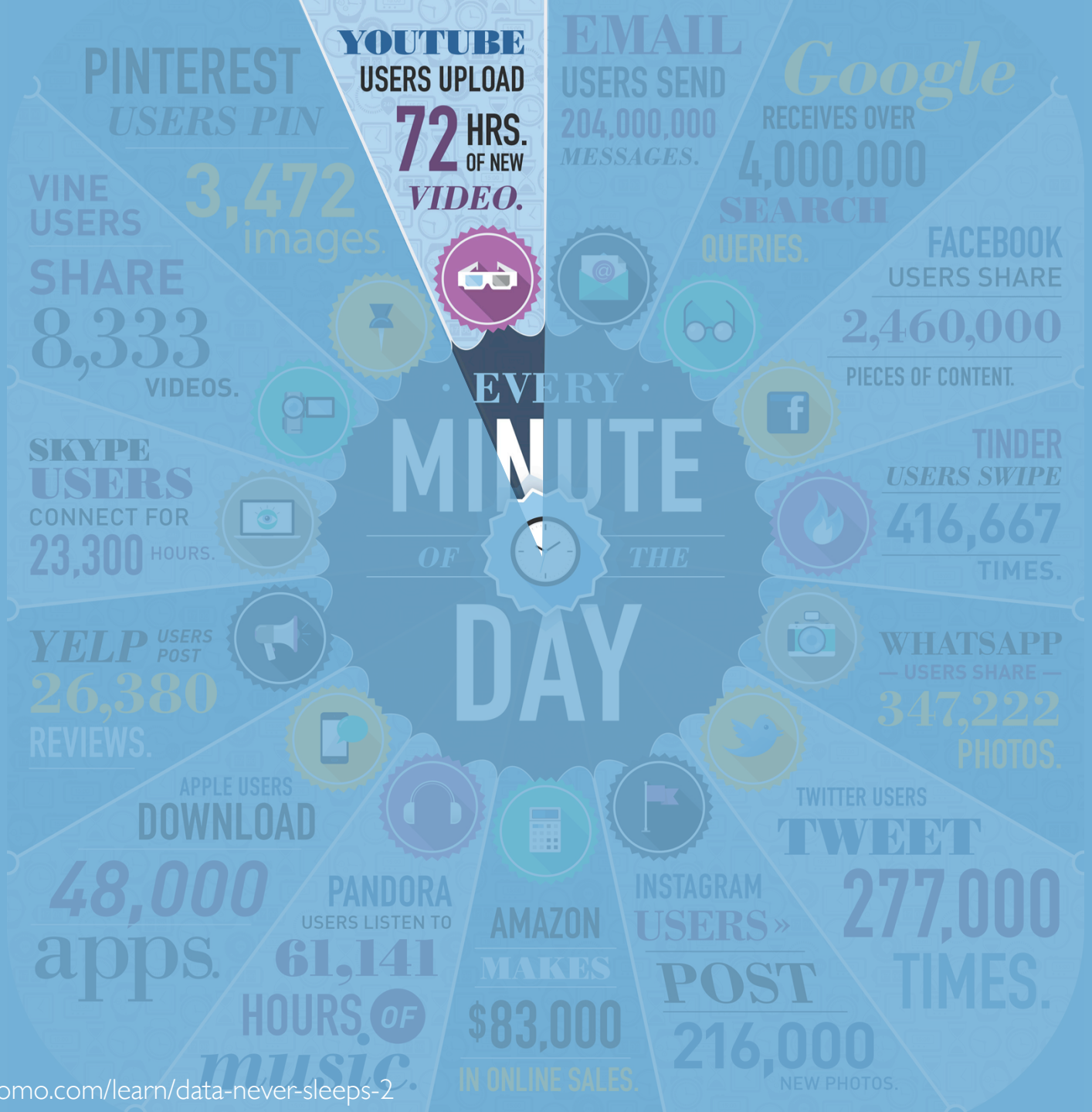
http://www.domo.com/learn/data-never-sleeps-2



http://www.domo.com/learn/data-never-sleeps-2



http://www.domo.com/learn/data-never-sleeps-2

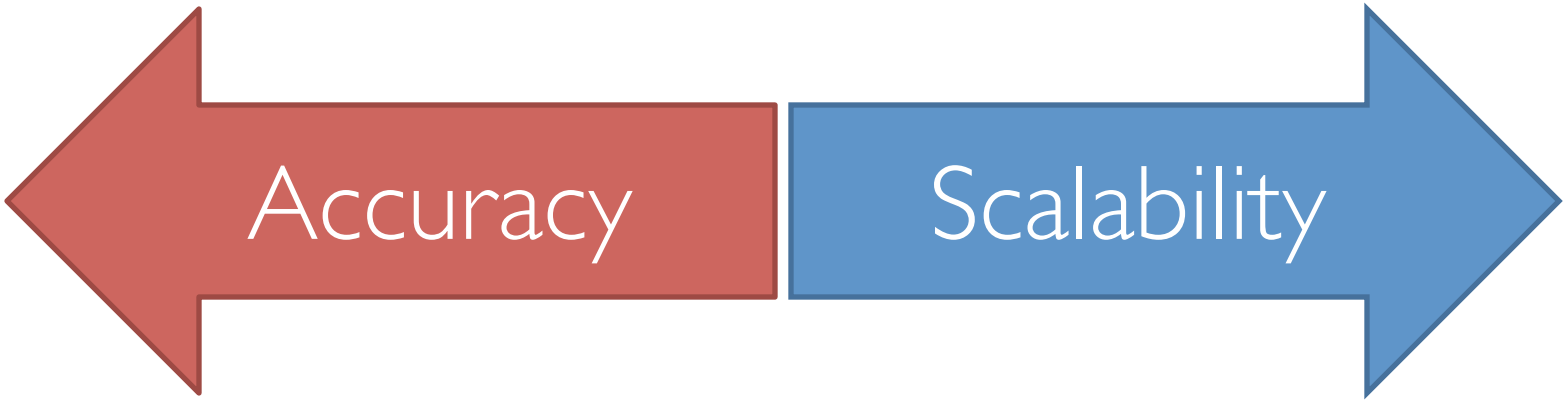


http://www.domo.com/learn/data-never-sleeps-2

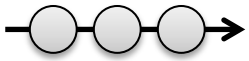
Data *Velocity*
is an *opportunity* for
Bayesian Nonparametrics.

How do we scale
Bayesian inference?

Opposing Forces

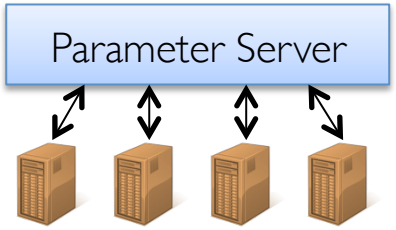


Ability to estimate the posterior distribution



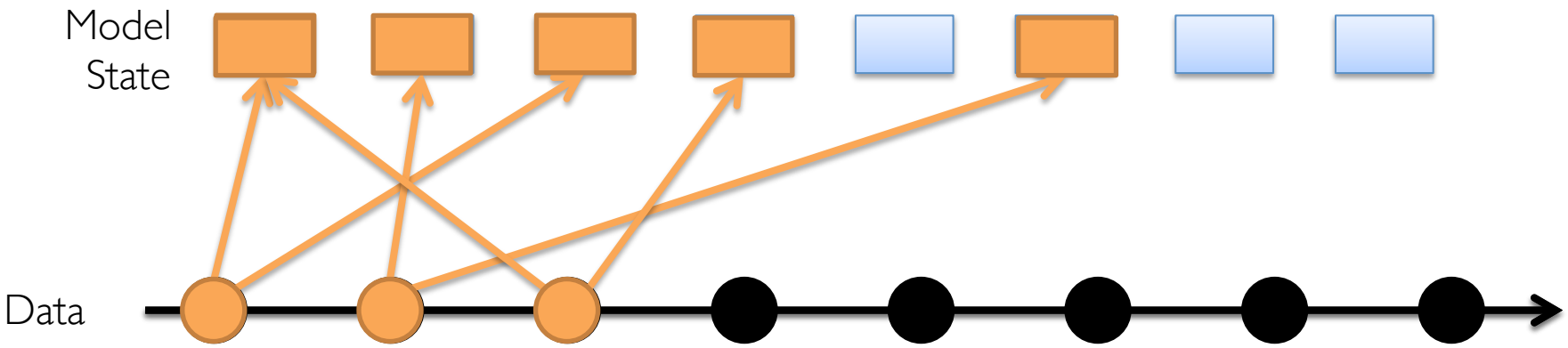
Serial Inference

Ability to effective use parallel resources

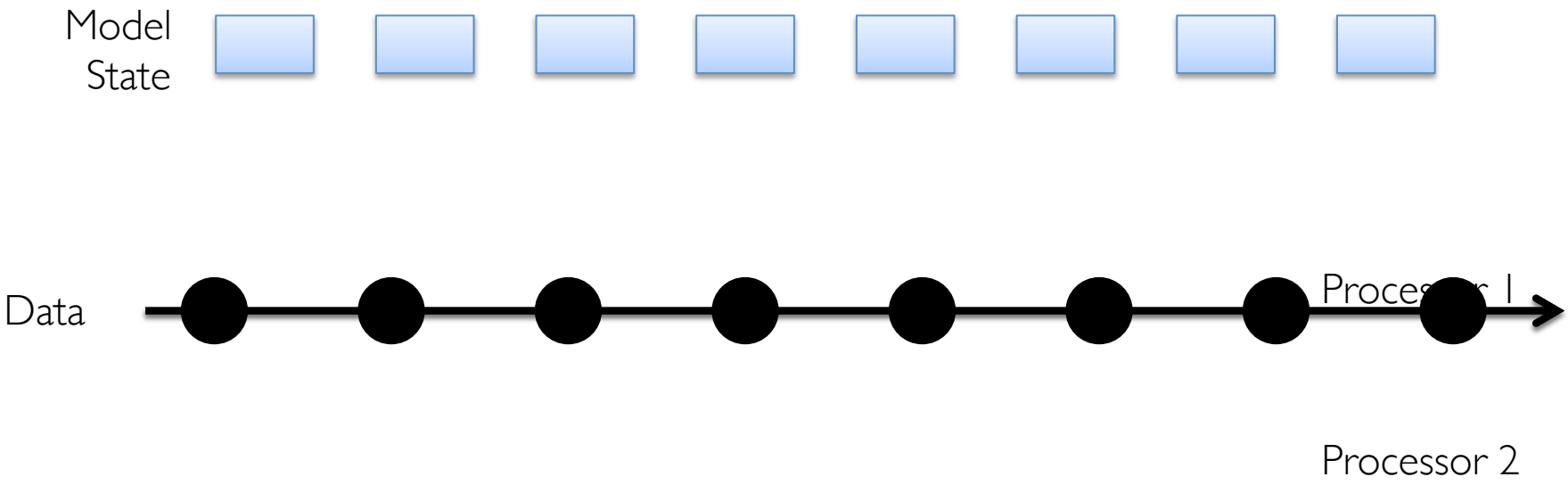


Coordination Free Samplers

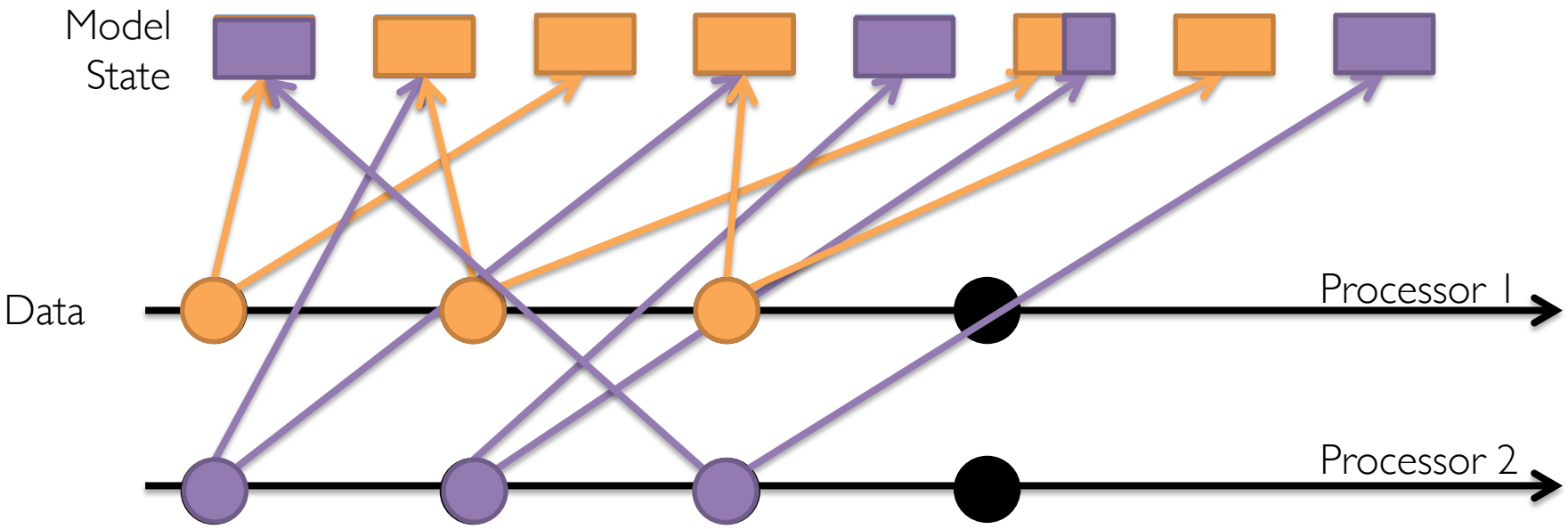
Serial Inference



Coordination Free Parallel Inference



Coordination Free Parallel Inference



Keep Calm and Carry On.

Parameter Servers

System for Coordination Free Inference

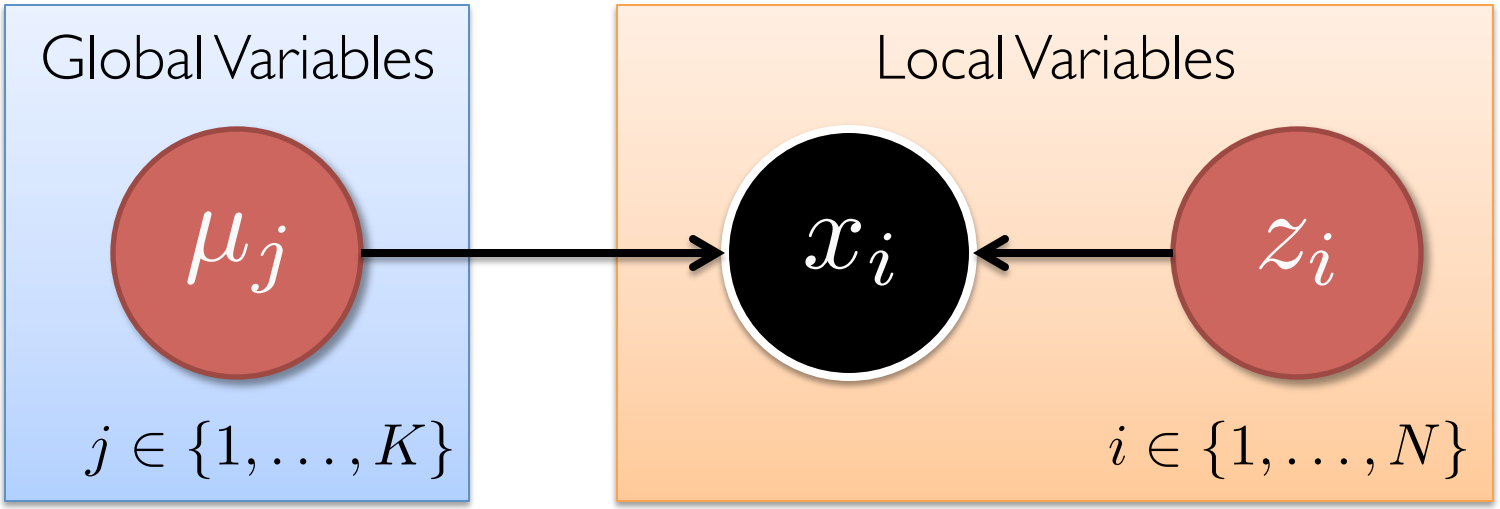
D. Newman, A. Asuncion, P. Smyth, and M. Welling. *Distributed inference for latent Dirichlet allocation.* In NIPS, 2007.

A. Smola and S. Narayanamurthy. *An architecture for parallel topic models.* VLDB'10

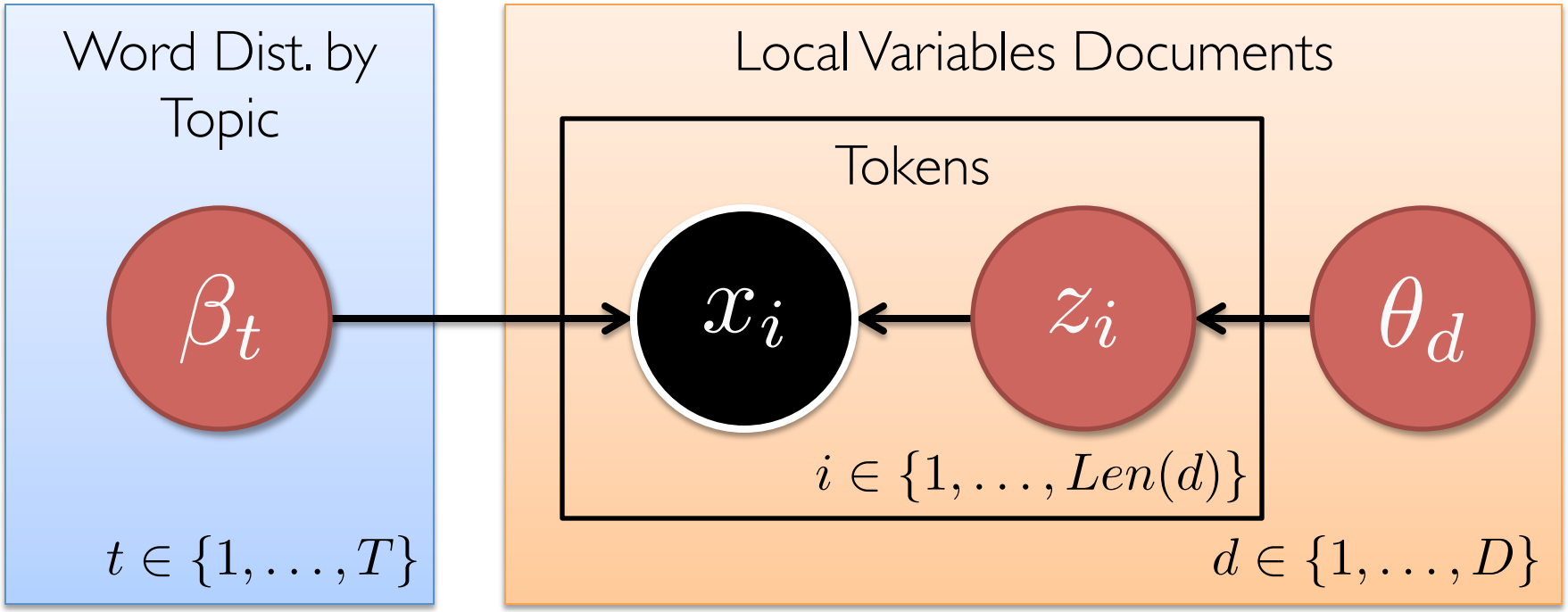
Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. *Scalable inference in latent variable models.* WSDM '12

Ho et al. "More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server." NIPS'13

Hierarchical Clustering



Example: Topic Modeling with LDA

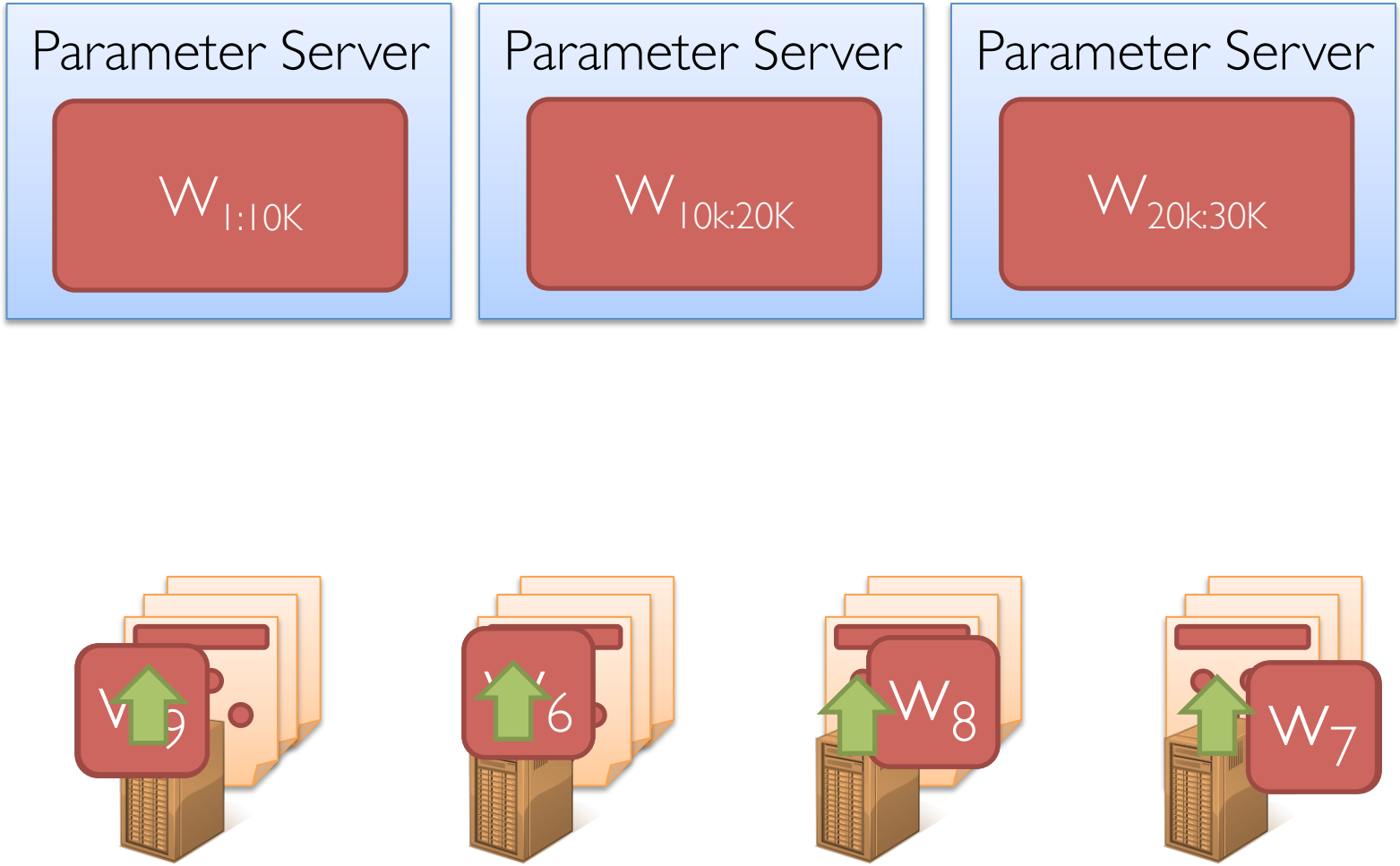


Maintained by the Parameter Server

Maintained by the Workers Nodes

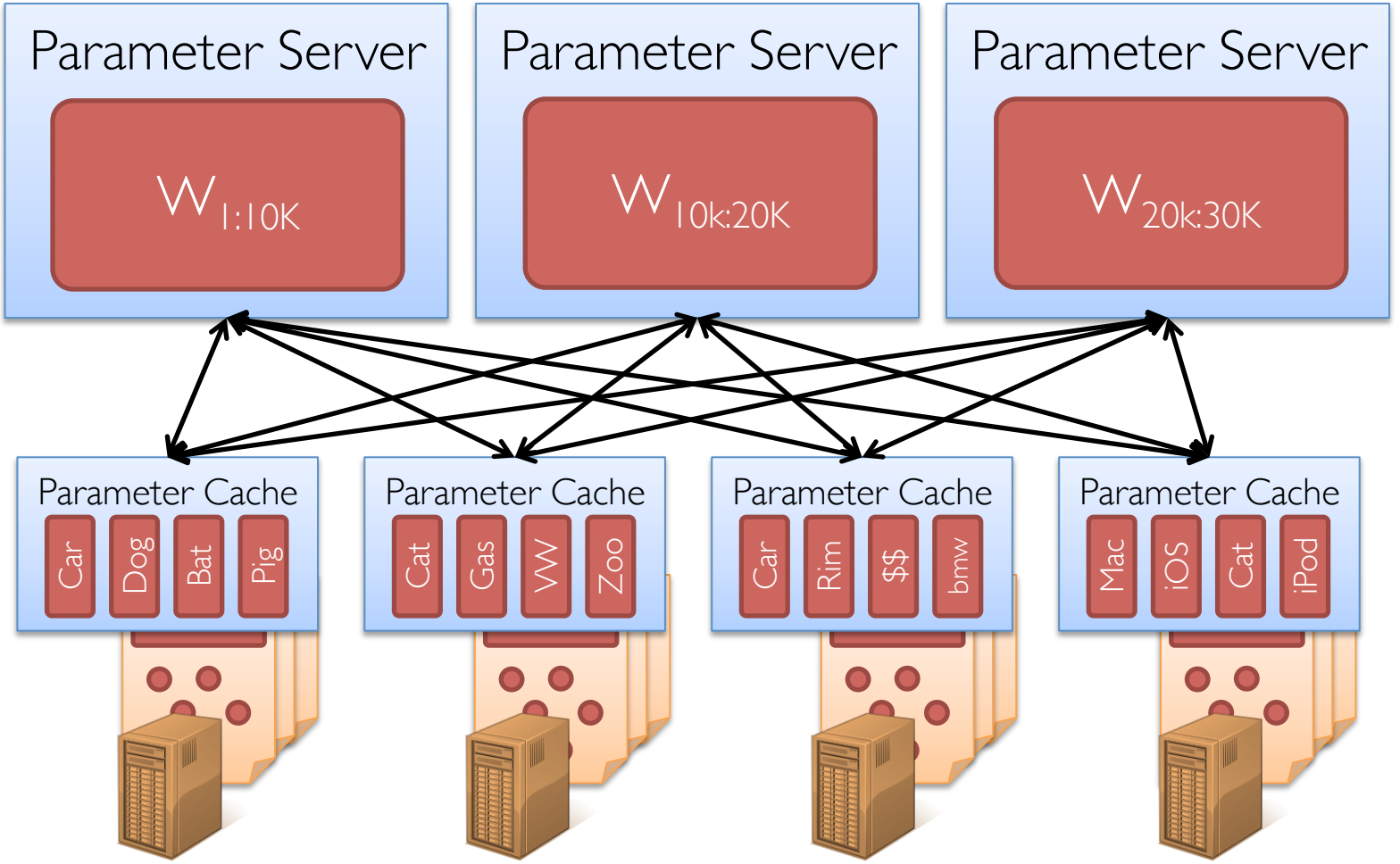
Ex: Collapsed Gibbs Sampler for LDA

Partitioning the model and data



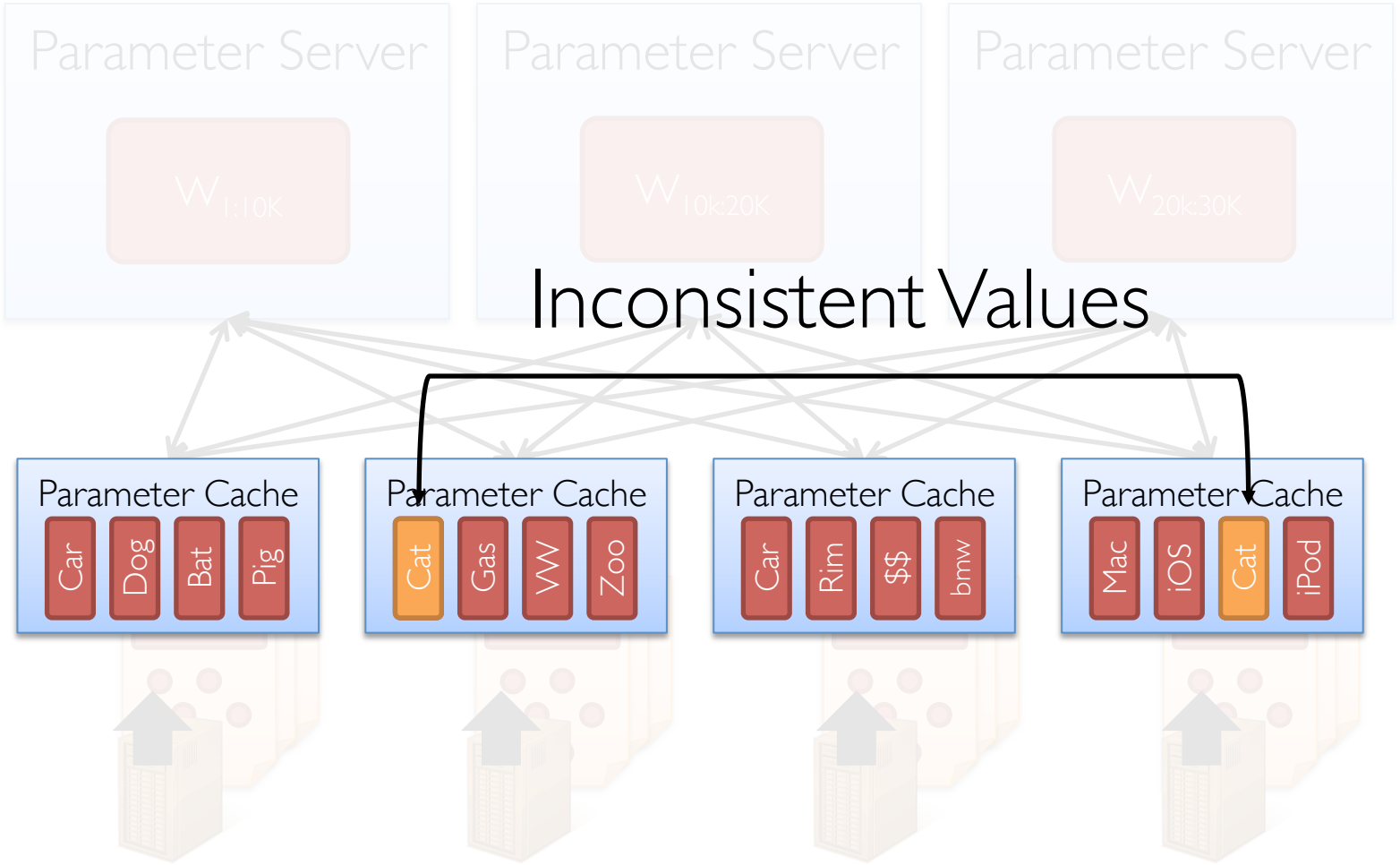
Ex: Collapsed Gibbs Sampler for LDA

Partitioning the model and data



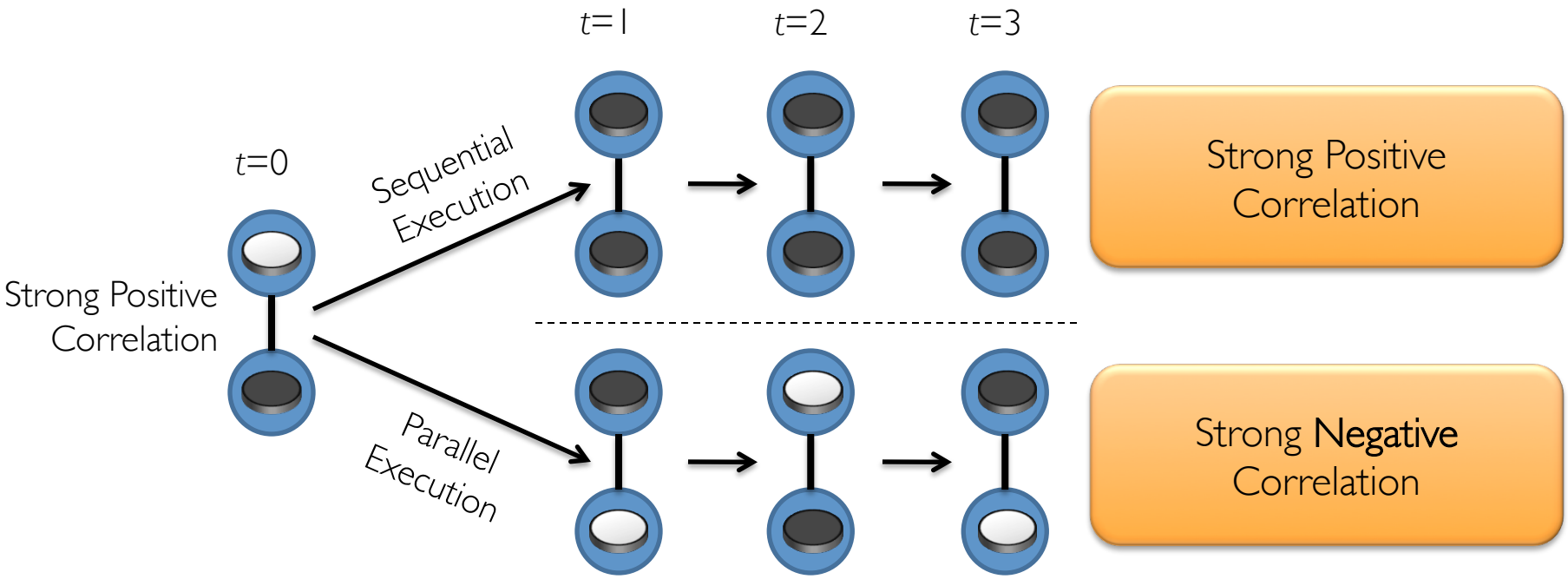
Ex: Collapsed Gibbs Sampler for LDA

Inconsistent model replicas



Parallel Gibbs Sampling

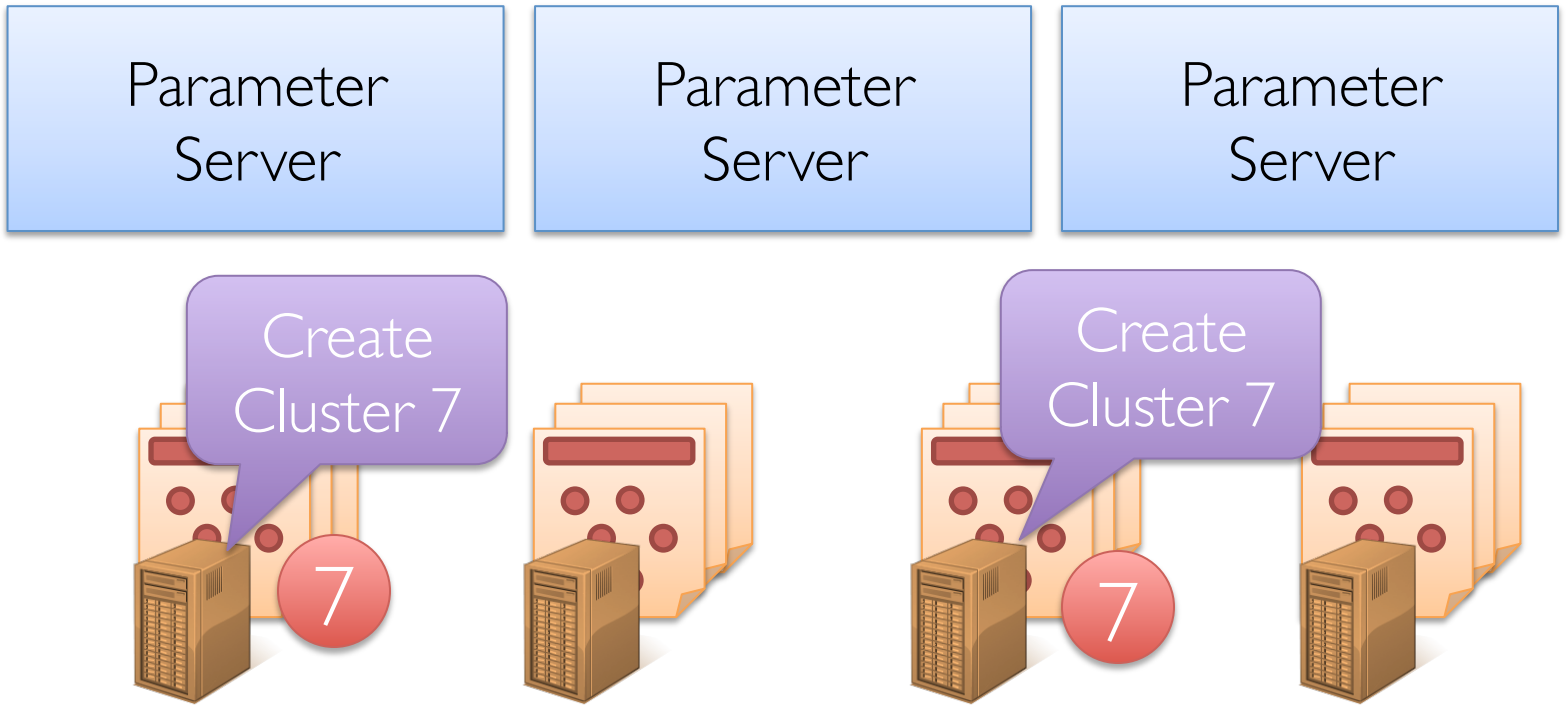
Incorrect Posterior



dependent variables ***cannot in general*** *be sampled* ***simultaneously.***

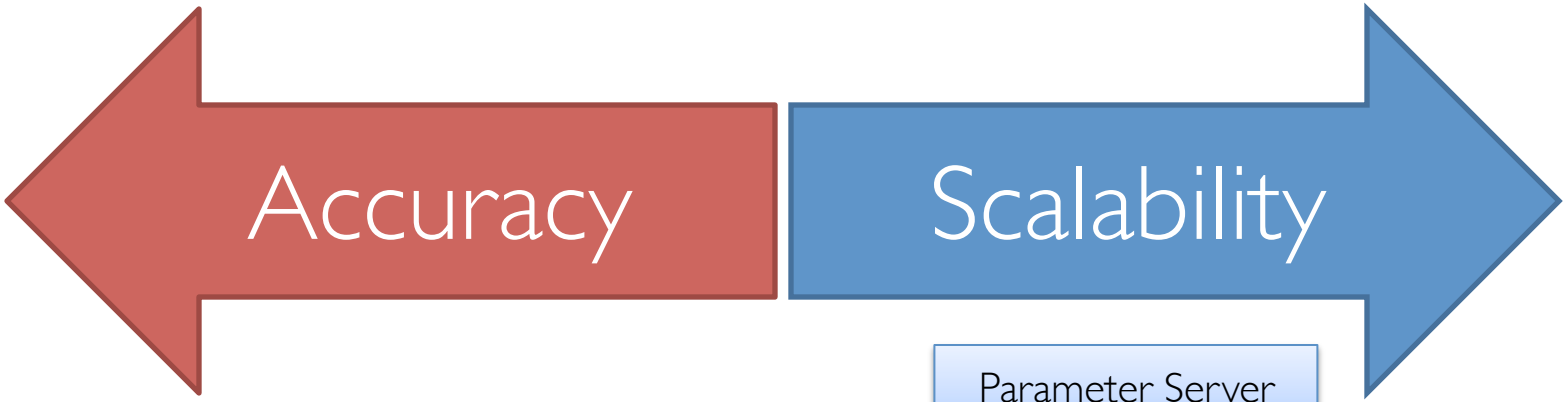
Issues with *Nonparametrics*

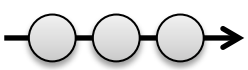
Difficult to introduce new clusters asynchronously:



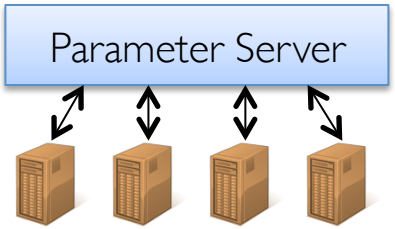
Leads to too many clusters!

Opposing Forces



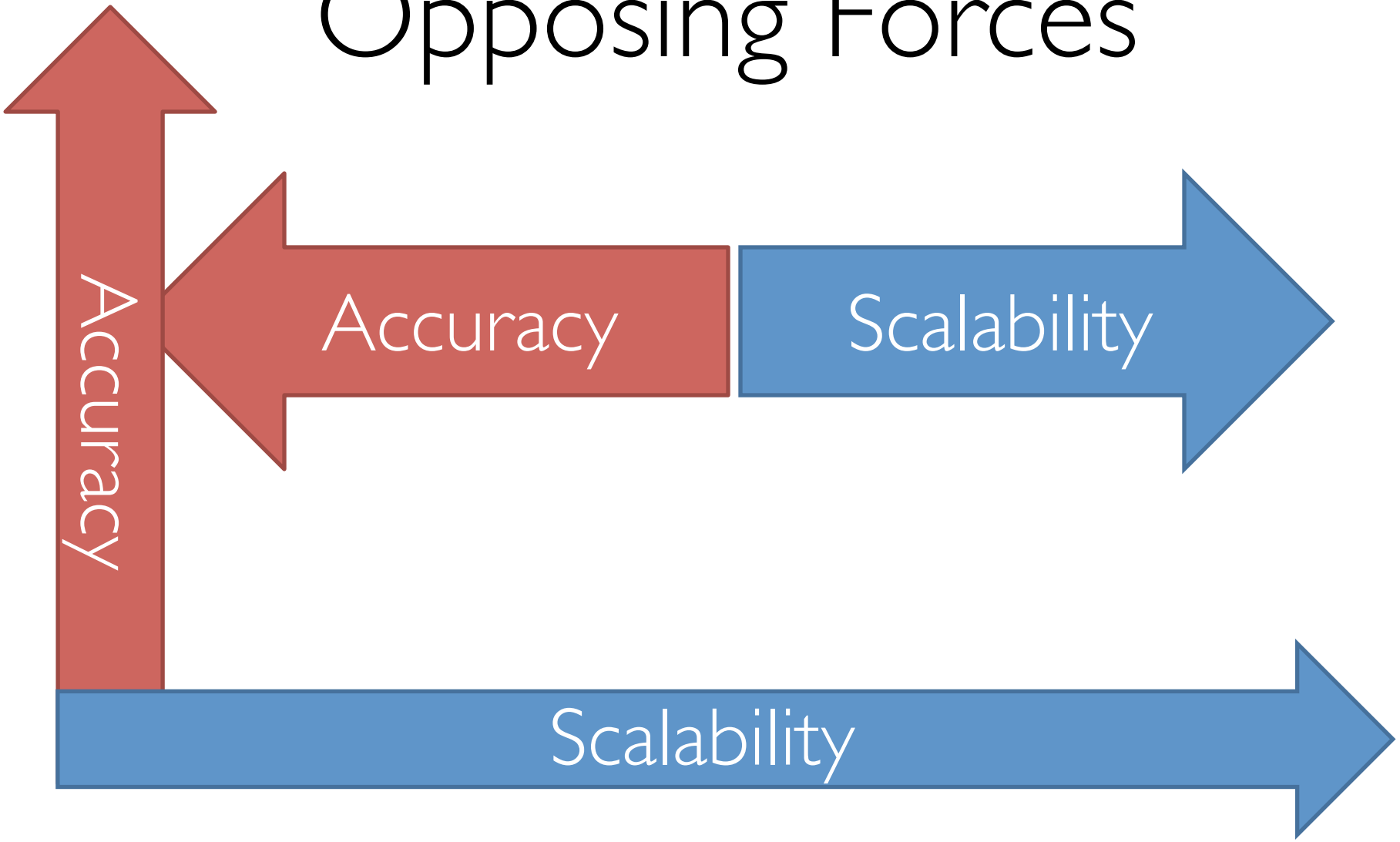


Serial
Inference

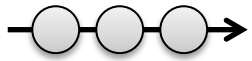


Asynchronous
Samplers

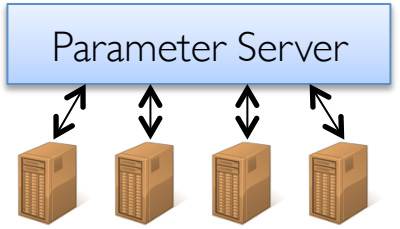
Opposing Forces



Opposing Forces



Serial
Inference

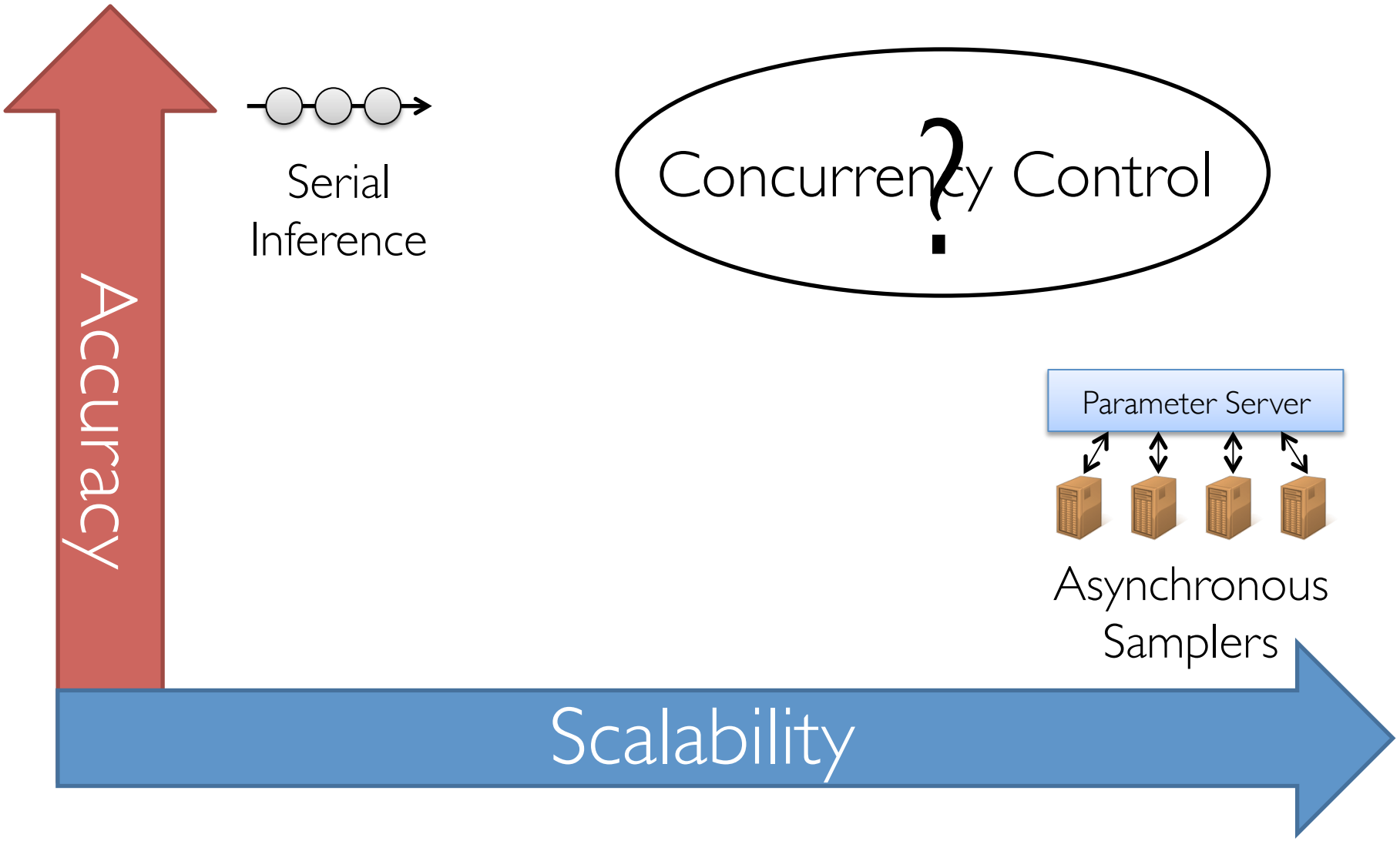


Asynchronous
Samplers

Accuracy

Scalability

Opposing Forces



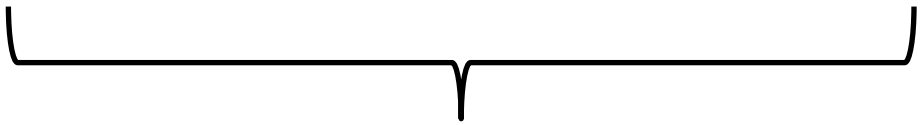
Concurrency Control

Coordination Free (Parameter Server):

Provably fast and correct under key assumptions.

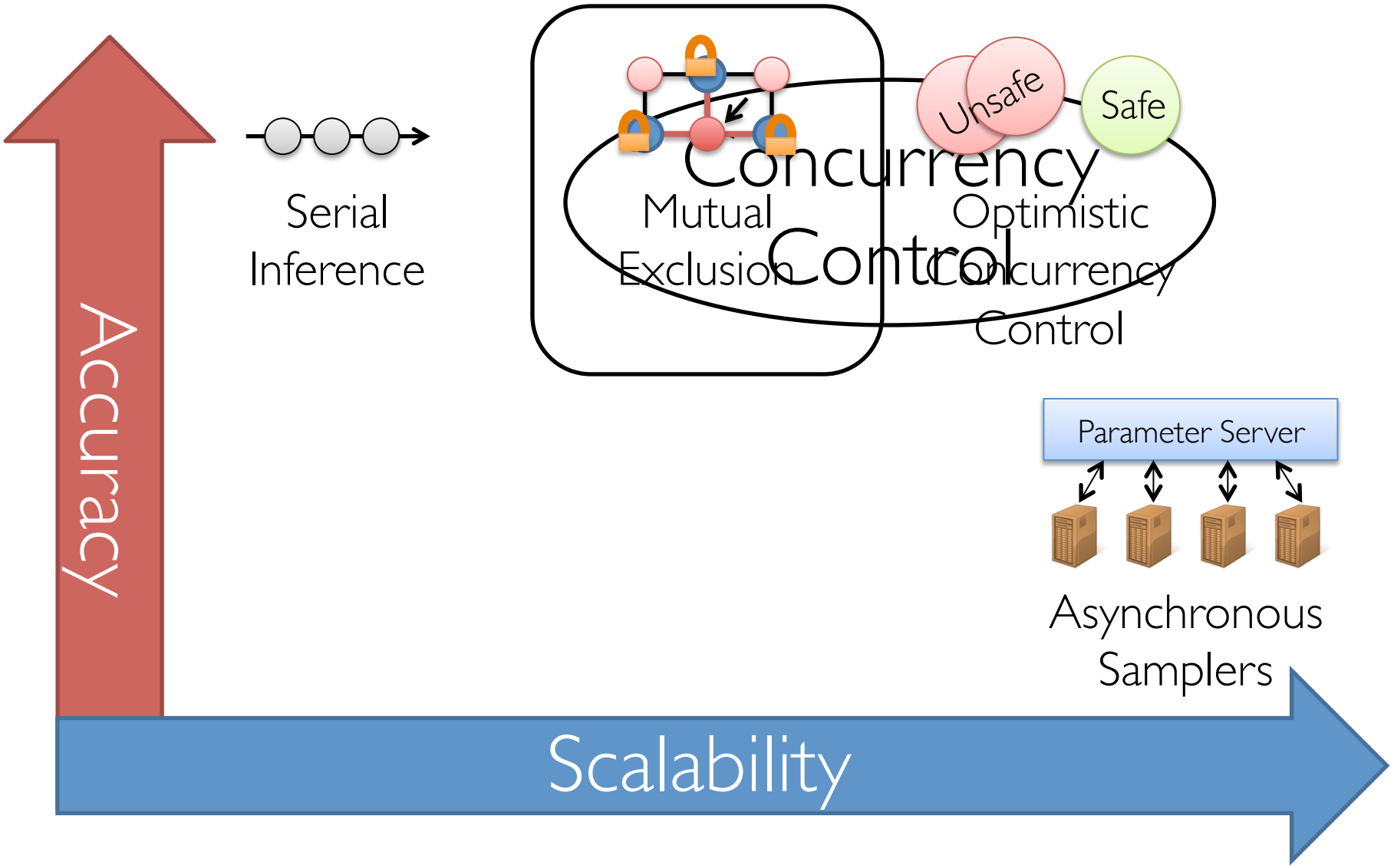
Concurrency Control:

Provably correct and fast under key assumptions.



Systems Ideas to
Improve Efficiency

Opposing Forces



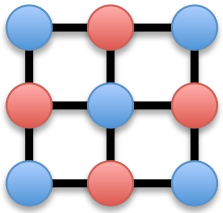
Mutual Exclusion

Conditional Independence

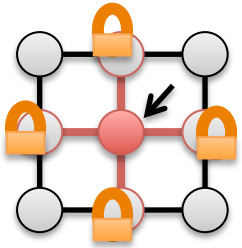
J. Gonzalez, Y. Low, A. Gretton, and C. Guestrin. *Parallel Gibbs Sampling: From Colored Fields to Thin Junction Trees*. AISTATS'11

Exploit the Markov random field for

Parallel Gibbs Sampling



Graph Coloring



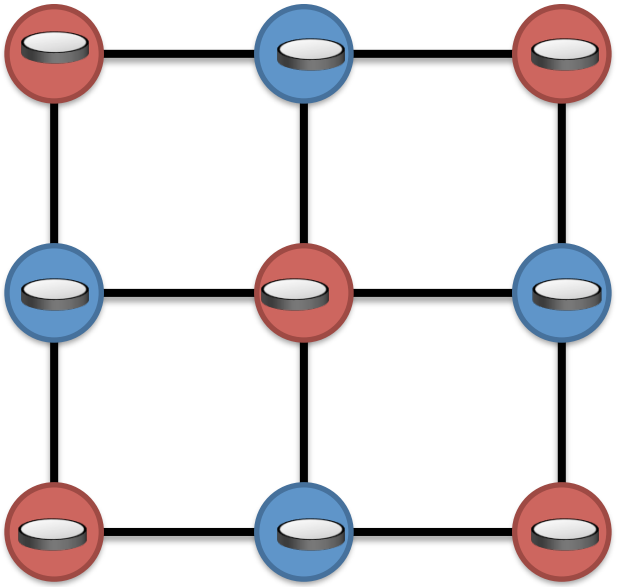
R/W Lock Mutual Exclusion

Mutual Exclusion through Scheduling

Chromatic Gibbs Sampler

Compute a k-coloring of the graphical model

Sample all variables with same color in parallel



Serial Equivalence:

Time

Theorem: *Chromatic* Sampler

Ergodic: converges to the correct distribution

» Based on graph coloring of the Markov Random Field

Quantifiable acceleration in **mixing**

Time to update all variables once

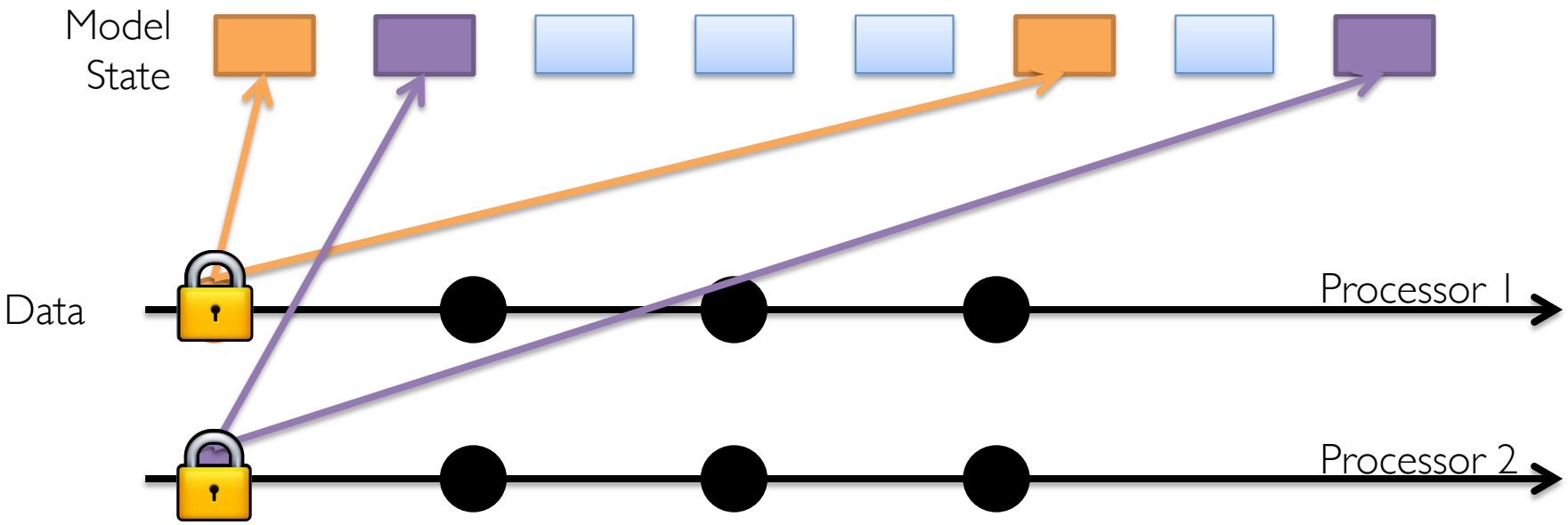
$$O\left(\frac{n}{p} + k\right)$$

Variables

Colors

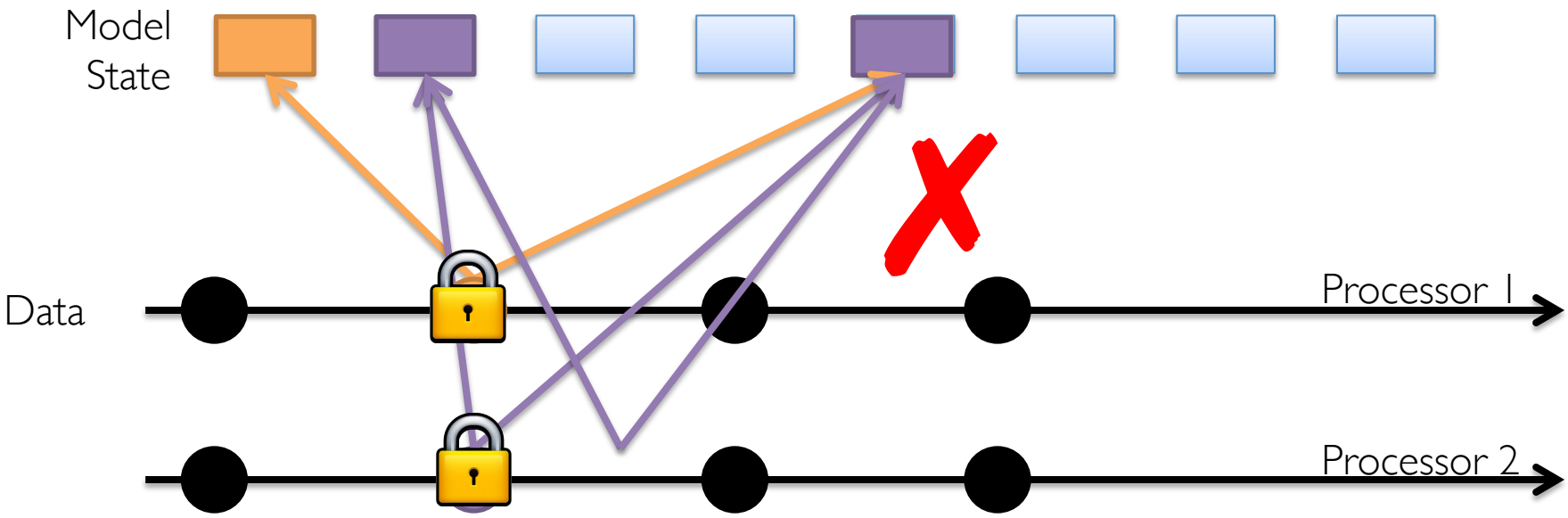
Processors

Mutual Exclusion Through Locking



Introducing locking (scheduling) protocols to identify potential conflicts.

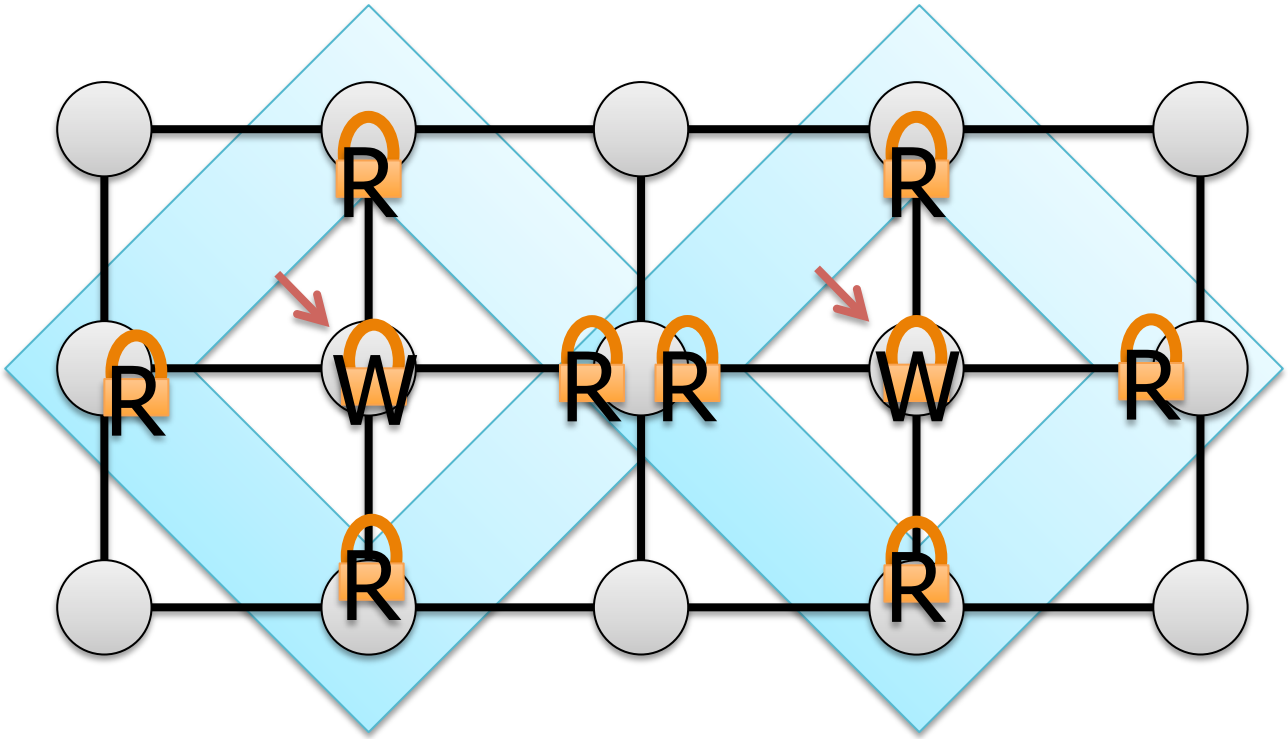
Mutual Exclusion Through Locking



Enforce serialization of computation that could conflict.

Markov Blanket Locks

Read/Write Locks:



Markov Blanket Locks

Eliminate fixed schedule and global coordination

Supports more advanced block sampling

Expected Parallelism:

$$\mathbf{E}(\#\text{active processors}) \geq 1 + (p-1)\left(1 - (p-1)\left(\frac{d+1}{n}\right)\right)$$

Max Degree

Processors

Variables

A System for Mutual Exclusion on Markov Random Fields

GraphLab/PowerGraph [UAI'10, OSDI'12]:

- Chromatic Sampling
- Markov Blanket Locks + Block Sampling

Limitation
Densely Connected MRF

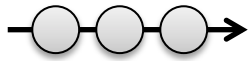
V-Structures: observations couple many variables

Collapsed models: clique-like MRFs

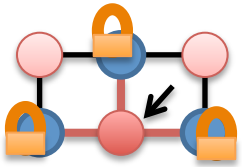
Mutual exclusion *pessimistically* serializes computation that *could* interfere.

Can we be *optimistic* and only serialize computation that *does* interfere?

Opposing Forces

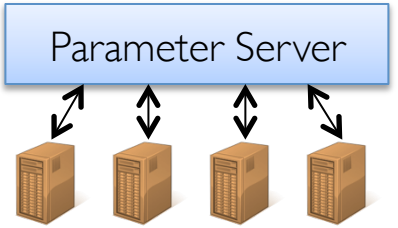


Serial Inference



Mutual Exclusion

Optimistic Concurrency Control



Asynchronous Samplers

Accuracy

Scalability

Optimistic Concurrency Control

assume the best and correct

X. Pan, J. Gonzalez, S. Jegelka, T. Broderick, M. Jordan. Optimistic Concurrency Control for Distributed Unsupervised Learning. NIPS'13

Xinghao Pan

Tamara Broderick

Stefanie Jegelka

Michael Jordan

Optimistic Concurrency Control

Classic idea from Database Systems:

Kung & Robinson. *On optimistic methods for concurrency control.* ACM Transactions on Database Systems. *1981*

Assume most operations won't conflict:

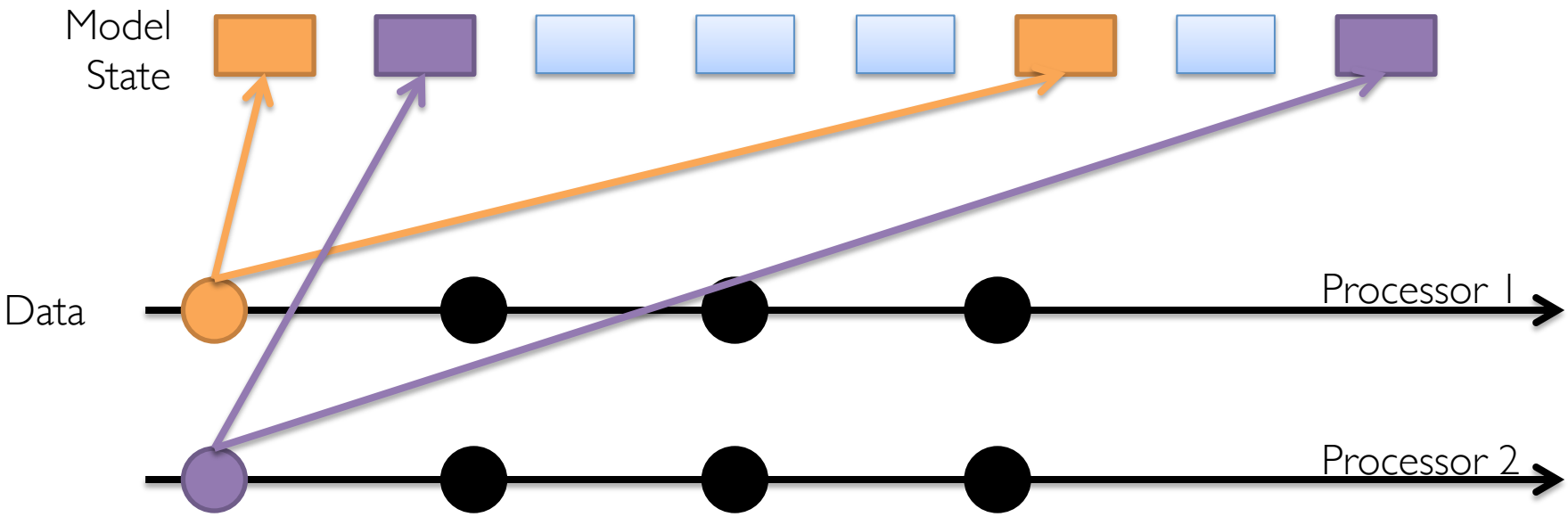
- Execute operations without blocking

  Frequent case is fast

- Identify and resolve conflicts *after* they occur

  Infrequent case with potentially costly resolution

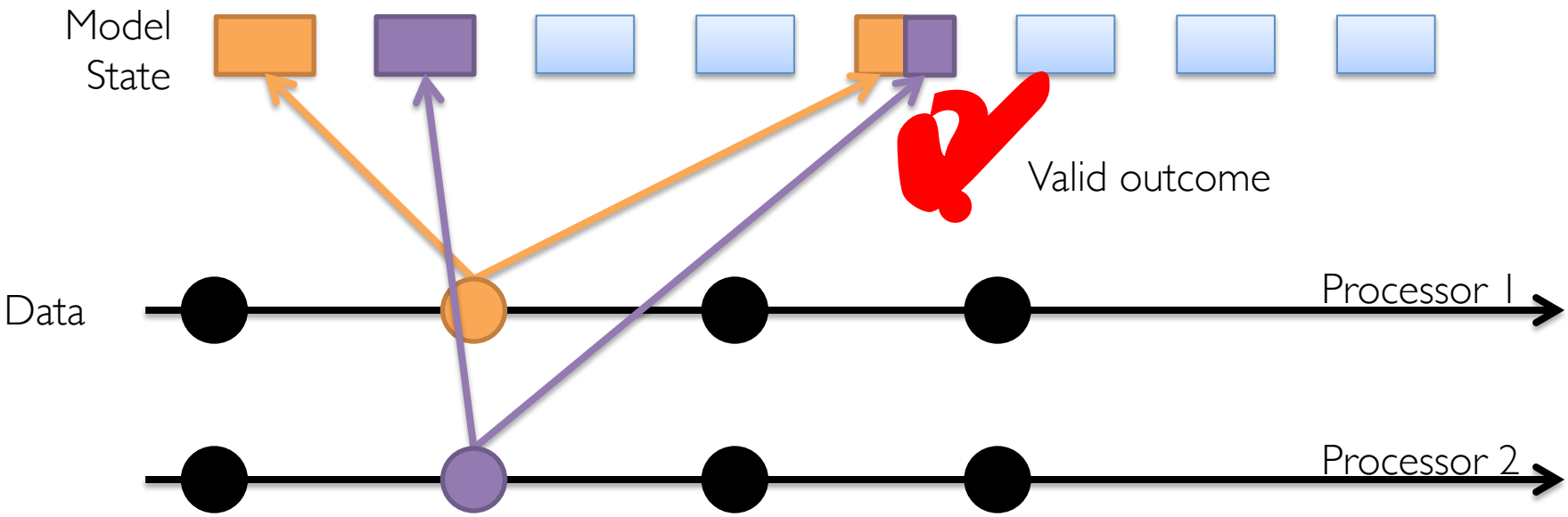
Optimistic Concurrency Control



Allow computation to proceed without blocking.

Kung & Robinson. *On optimistic methods for concurrency control.*
ACM Transactions on Database Systems 1981

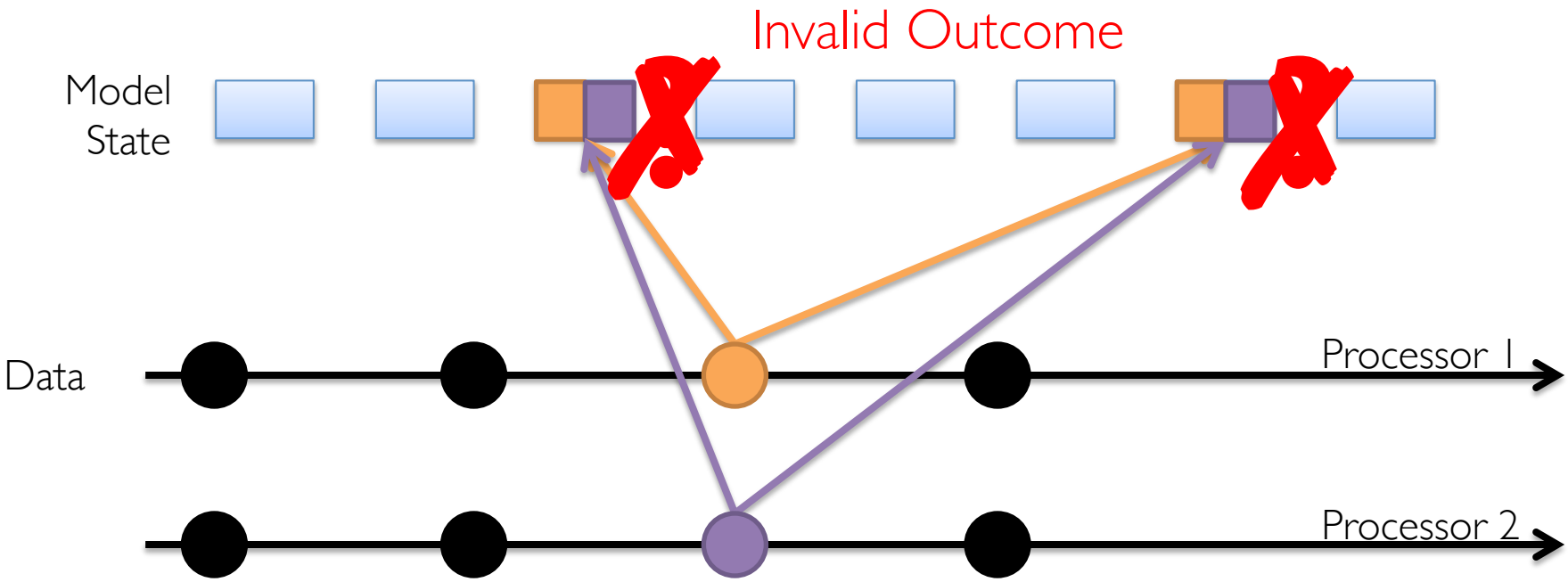
Optimistic Concurrency Control



Validate potential conflicts.

Kung & Robinson. *On optimistic methods for concurrency control.*
ACM Transactions on Database Systems 1981

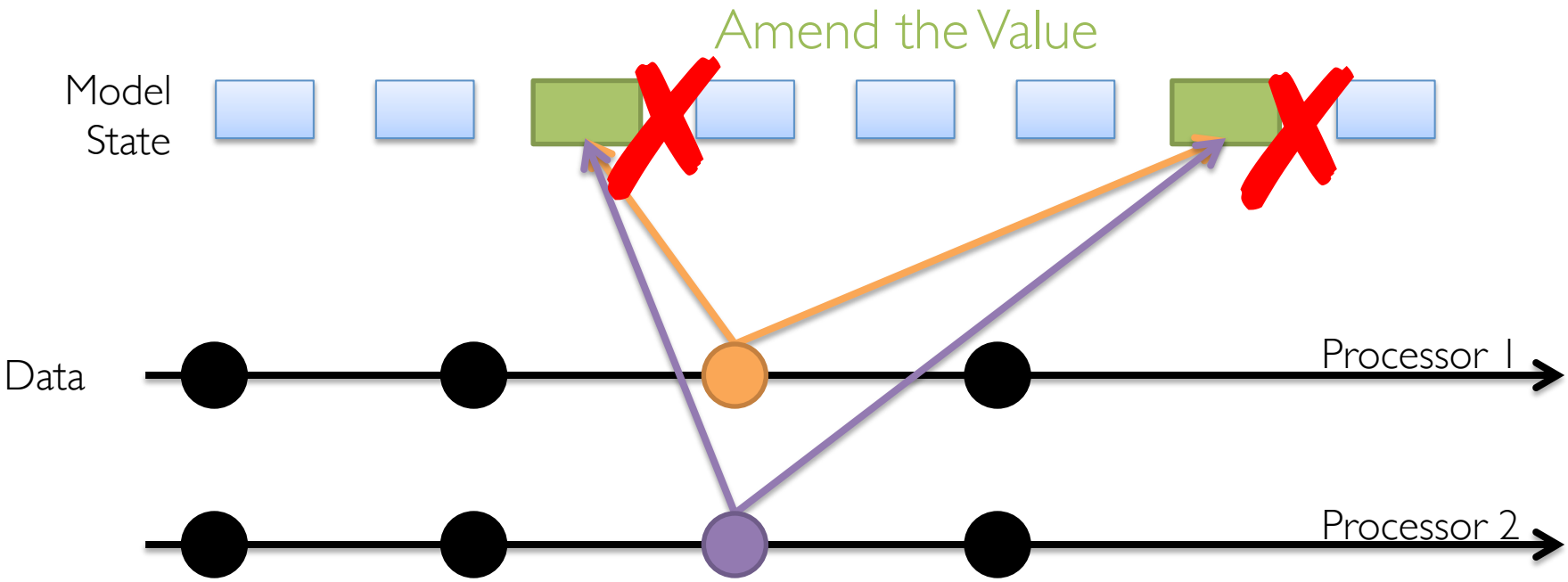
Optimistic Concurrency Control



Validate potential conflicts.

Kung & Robinson. *On optimistic methods for concurrency control.* ACM Transactions on Database Systems 1981

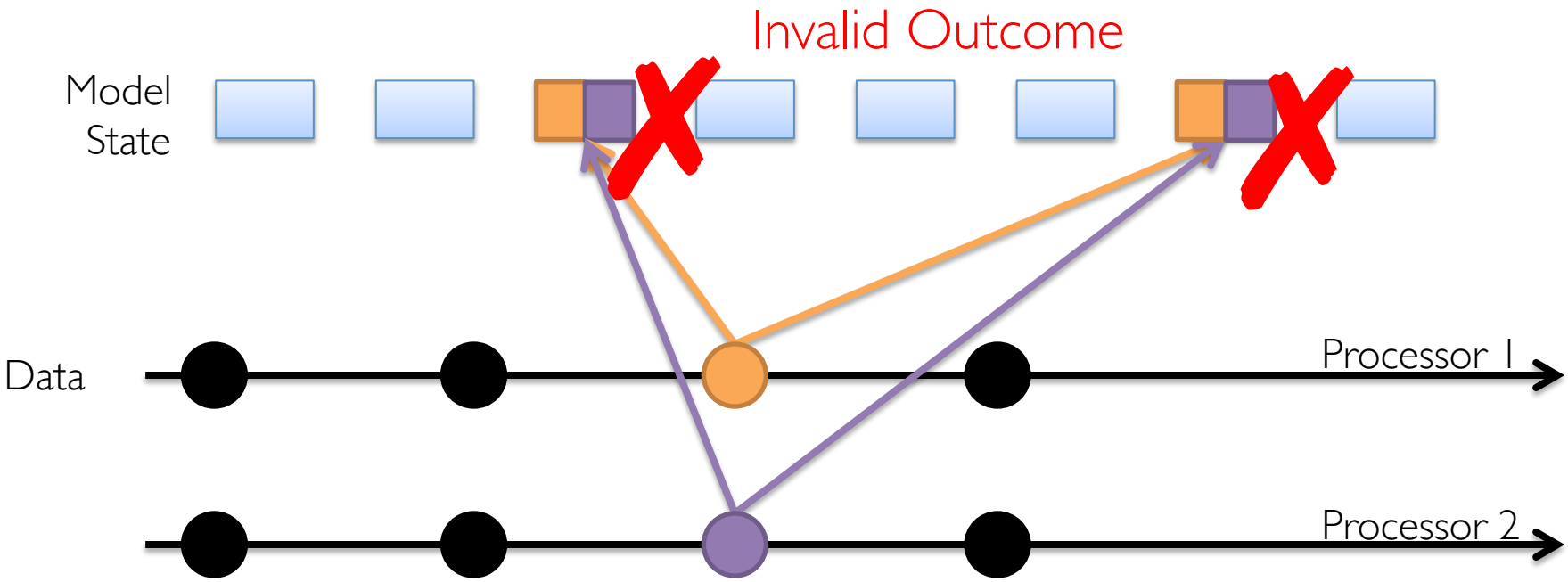
Optimistic Concurrency Control



Take a compensating action.

Kung & Robinson. *On optimistic methods for concurrency control.* ACM Transactions on Database Systems 1981

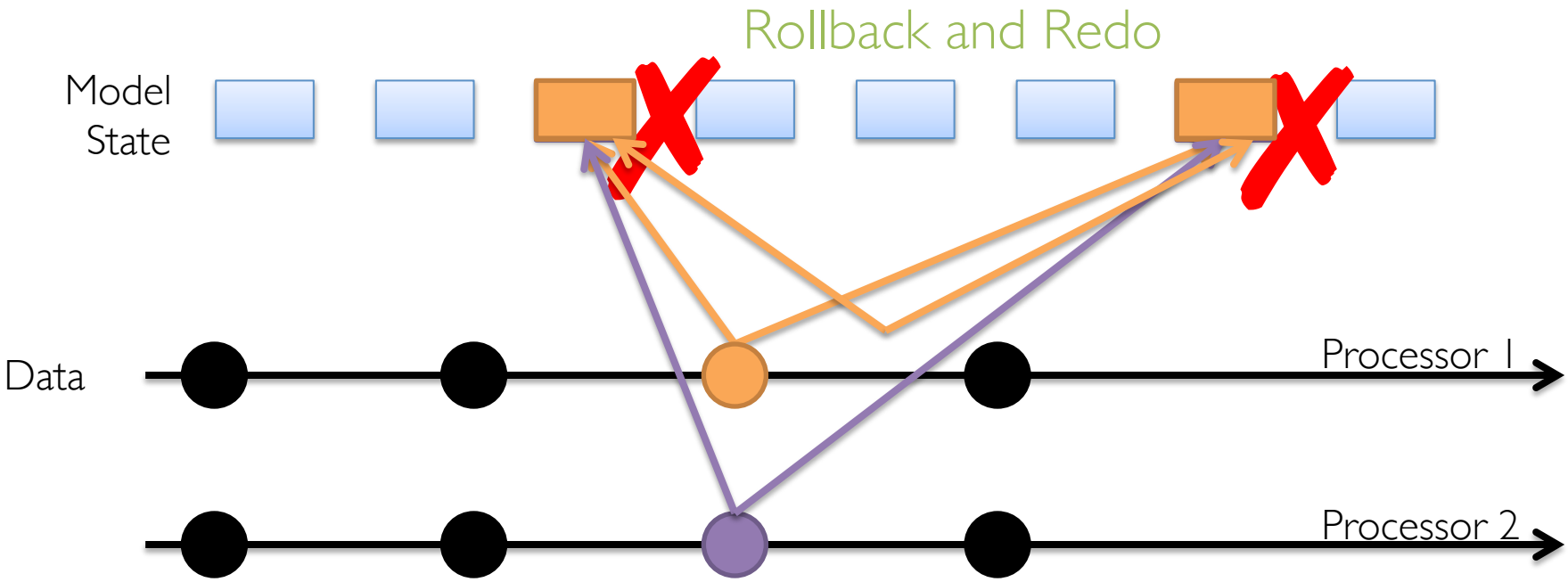
Optimistic Concurrency Control



Validate potential conflicts.

Kung & Robinson. *On optimistic methods for concurrency control.* ACM Transactions on Database Systems 1981

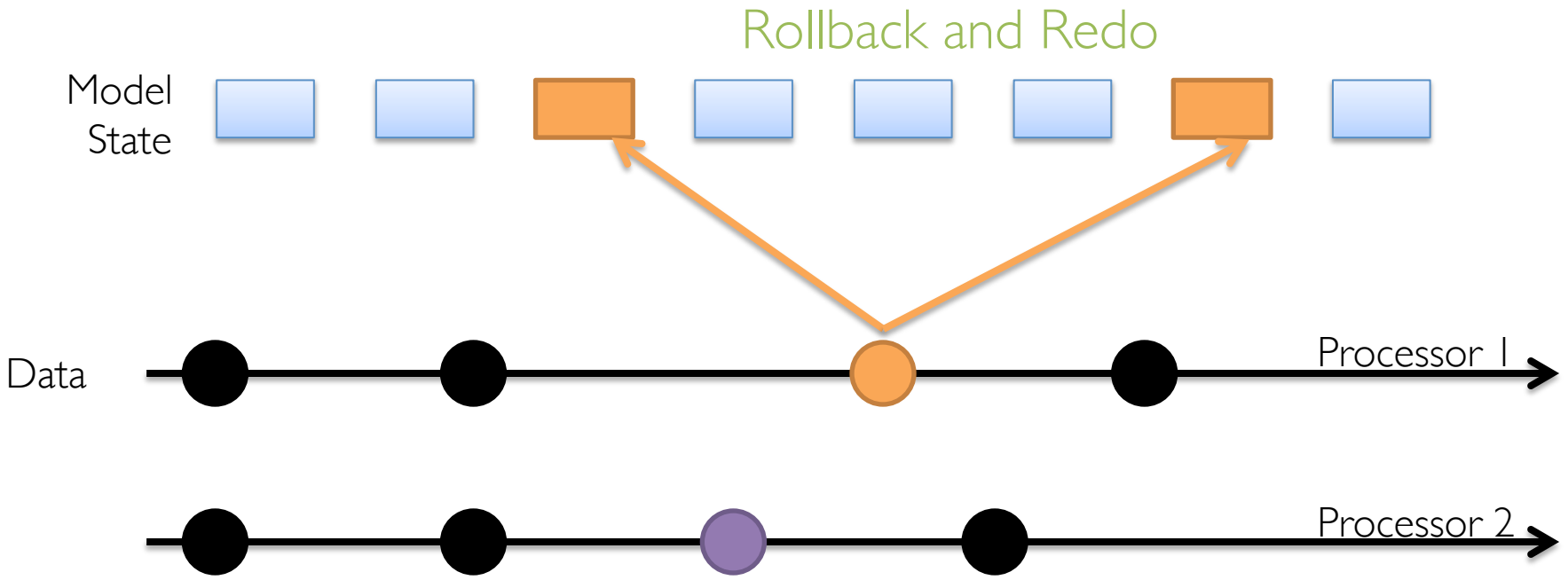
Optimistic Concurrency Control



Take a compensating action.

Kung & Robinson. *On optimistic methods for concurrency control.* ACM Transactions on Database Systems 1981

Optimistic Concurrency Control



Non-Blocking Computation ➡ Concurrency

Requirements:

Fast → Validation: *Identify Errors*

Infrequent → Resolution: *Correct Errors*

➡ Accuracy

Optimistic Concurrency Control for Bayesian Inference

Non-parametric Models [Pan et al., NIPS'13]:

- OCC DP-Means: Dirichlet Process Clustering
- OCC BP-Means: Beta Process Feature Learning

Conditional Sampling: (*In Progress*)

- Collapsed Gibbs LDA
- Retrospective Sampling for HDP

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

Start with DP Gaussian mixture model:

$$G \sim \mathrm{DP}\left(\alpha, G_0 = \mathcal{N}(0, \rho I)\right)$$

$$\phi_i \sim G$$

$$x_i \sim \mathcal{N}\left(\phi_i, \sigma I\right)$$

small variance limit $\sigma \to 0$

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

Start with DP Gaussian mixture model:

$$G \sim \mathrm{DP}\left(\alpha, G_0 = \mathcal{N}(0, \rho I)\right)$$

$$\phi_i \sim G$$

$$x_i \sim \mathcal{N}\left(\phi_i, \sigma I\right)$$

small variance limit $\sigma \to 0$ redefine α:

$$\alpha(\sigma) = \left(1 + \frac{\rho}{\sigma}\right)^{d/2} \exp\left(-\frac{\lambda}{2\sigma}\right)$$

Decreases Rapidly

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

Corresponding Gibbs sampler conditionals:

$$\mathbf{P}\left(\text{join } c\right) \propto n_{-i,c} \exp\left(-\frac{||x_i - \mu_c||^2}{2\sigma}\right)$$

$$\mathbf{P}\left(\text{new}\right) \propto \exp\left(-\frac{1}{2\sigma}\left[\lambda + \frac{\sigma}{\rho + \sigma}||x_i||^2\right]\right)$$

Taking the small variance limit $\sigma \to 0$

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

Gibbs updates become deterministic:

$$
\begin{array}{l}
\textbf{for} \;\; i \in \{1, \ldots, n\} \;\textbf{do} \\
\quad c_i^{\min} = \arg\min_c ||x_i - \mu_c|| \\
\quad \textbf{if} \;\; \left|\left|x_i - \mu_{c_i^{min}}\right|\right| < \lambda \;\textbf{then} \;\; \text{join } c_i^{\min} \\
\quad \textbf{else} \;\; \text{create new cluster at } \mu_{k+1} = x_i
\end{array}
$$

Taking the small variance limit $\sigma \rightarrow 0$

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

Gibbs updates become deterministic:

for $i \in \{1, \ldots, n\}$ **do**

$c_i^{\min} = \arg\min_c ||x_i - \mu_c||$

if $\left|\left|x_i - \mu_{c_i^{min}}\right|\right| < \lambda$ **then** join $c_i^{\min}$

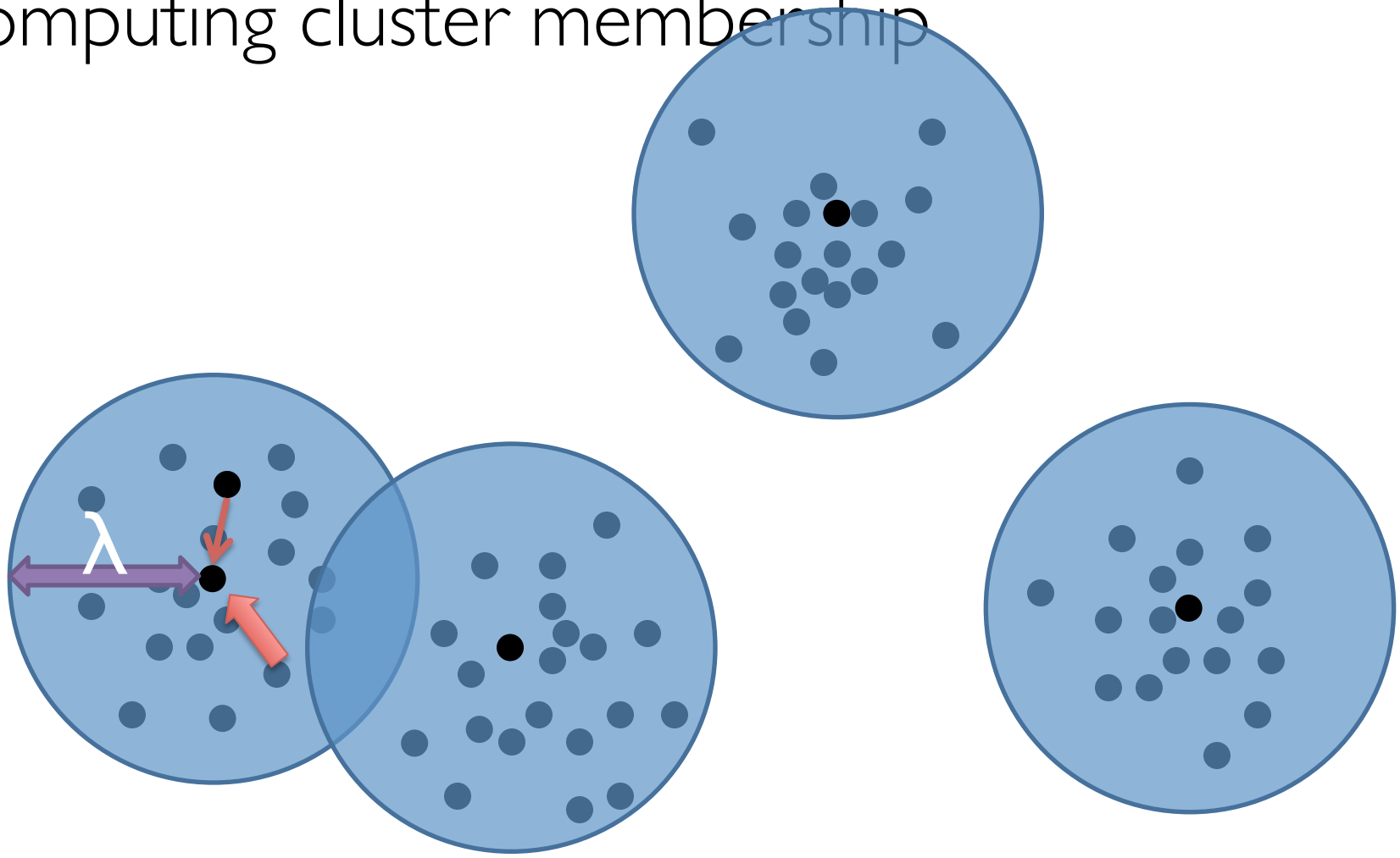
else create new cluster at $\mu_{k+1} = x_i$

for c *in clusters* **do** $\mu_c \leftarrow \frac{1}{n_c} \sum_{x \in c} x$

DP-Means Algorithm

[Kulis and Jordan, ICML'12]

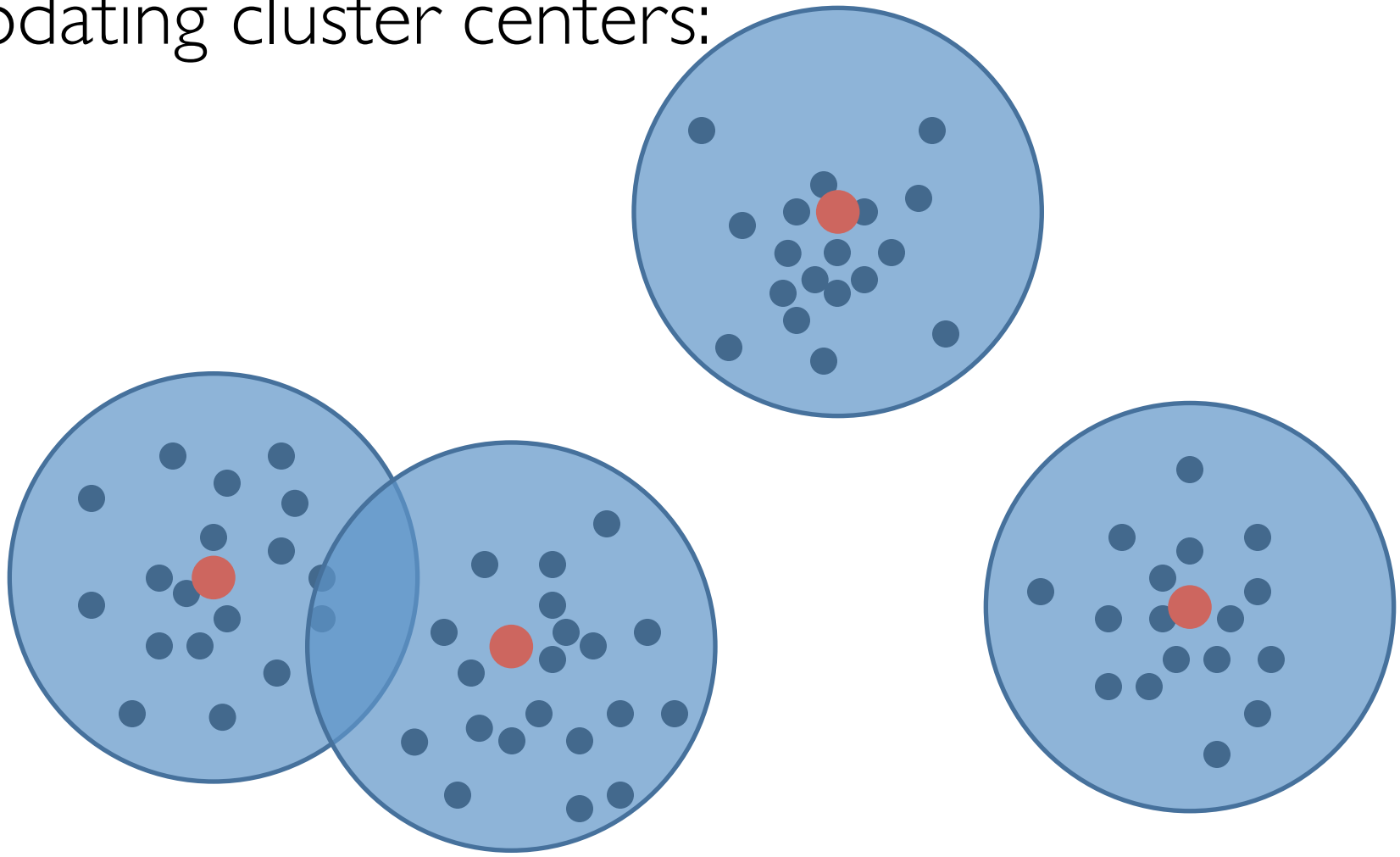
Computing cluster membership



DP-Means Algorithm

[Kulis and Jordan, ICML'12]

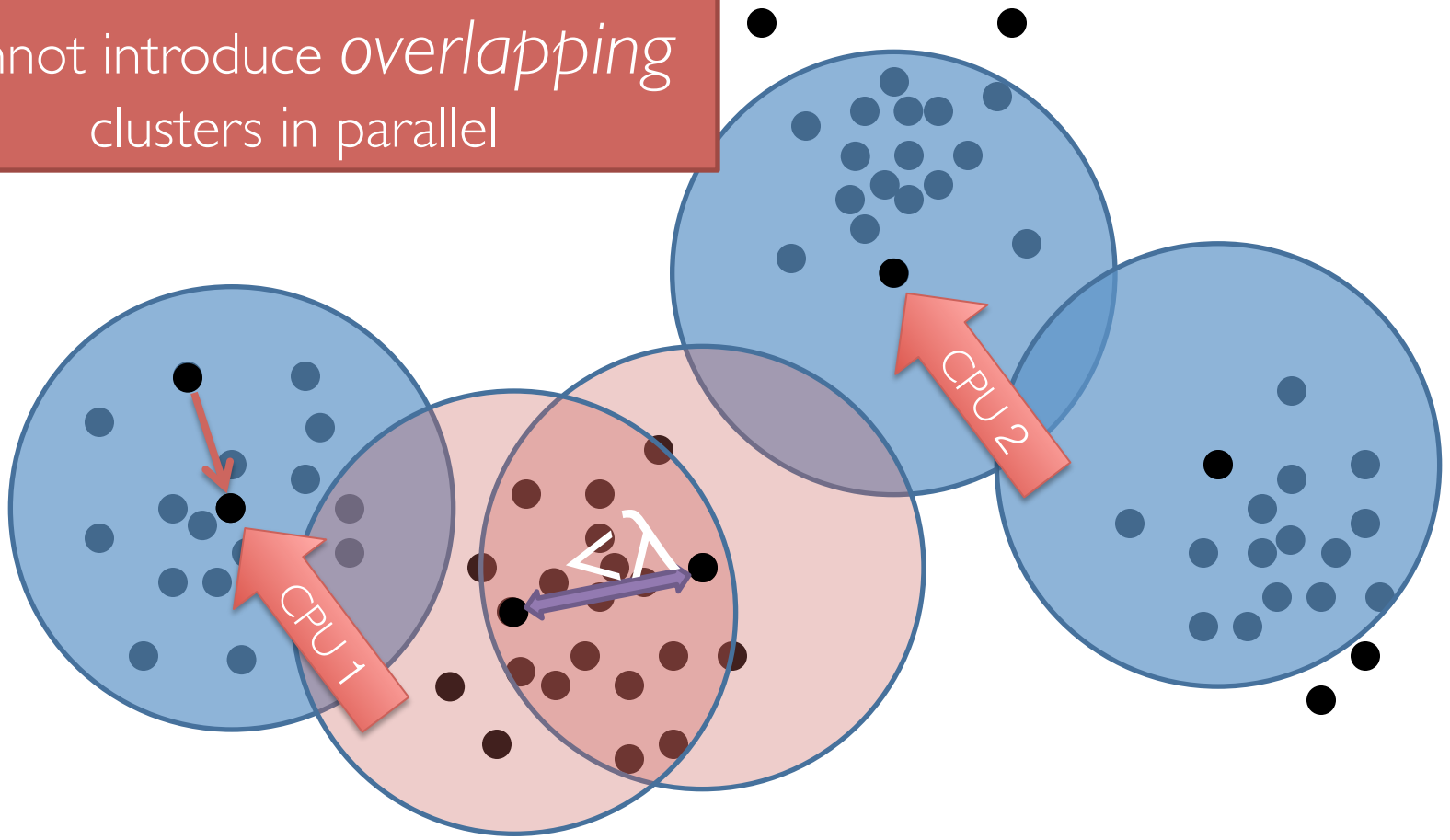
Updating cluster centers:



DP-Means Parallel Execution

Computing cluster membership in parallel:

Cannot introduce *overlapping* clusters in parallel



Optimistic Concurrency Control for Parallel DP-Means

Optimistic Assumption

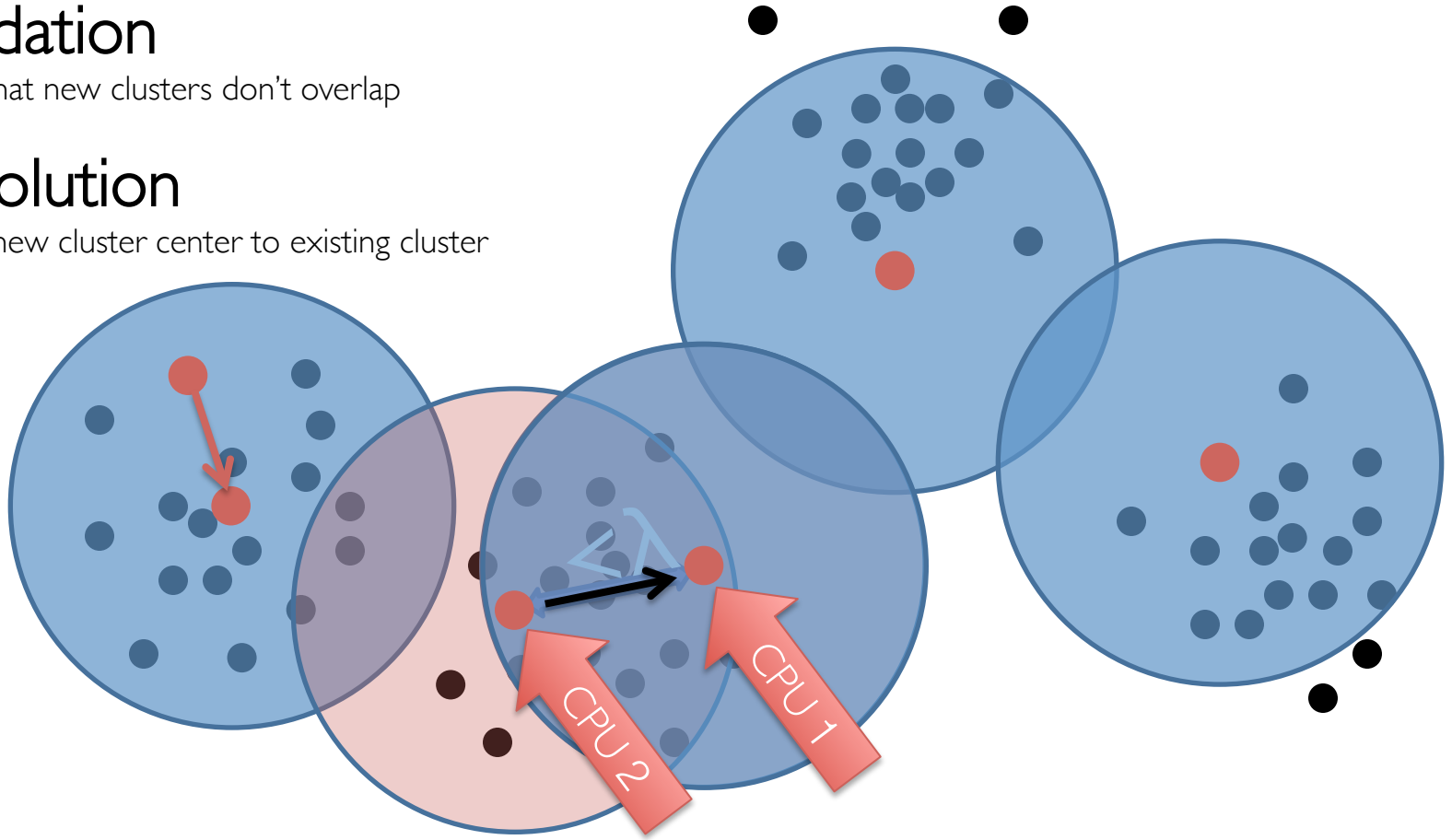
No new cluster created nearby

Validation

Verify that new clusters don't overlap

Resolution

Assign new cluster center to existing cluster



OCC DP-means

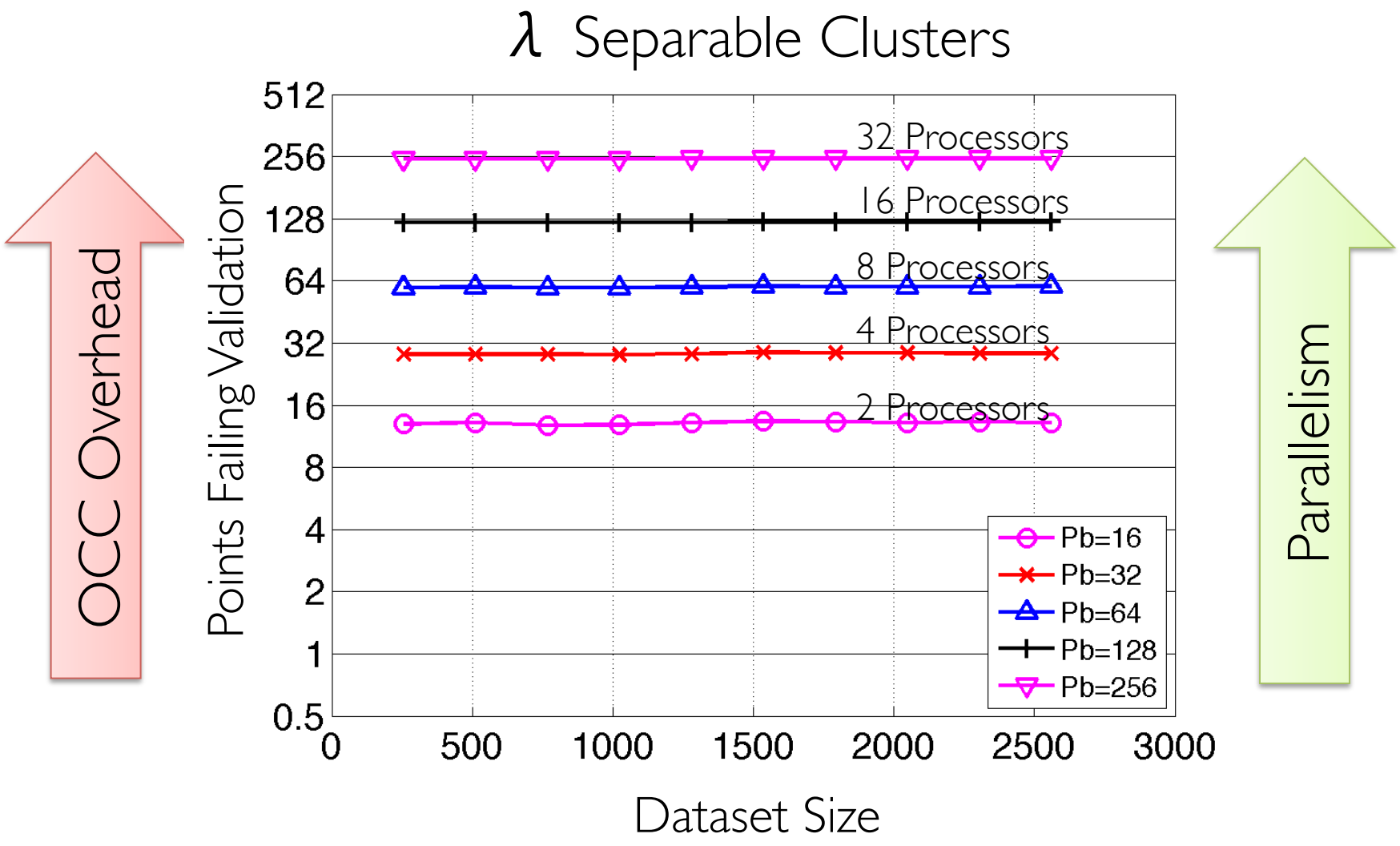
Theorem: OCC DP-means is serializable and therefore preserves theoretical properties of DP-means.

Correctness

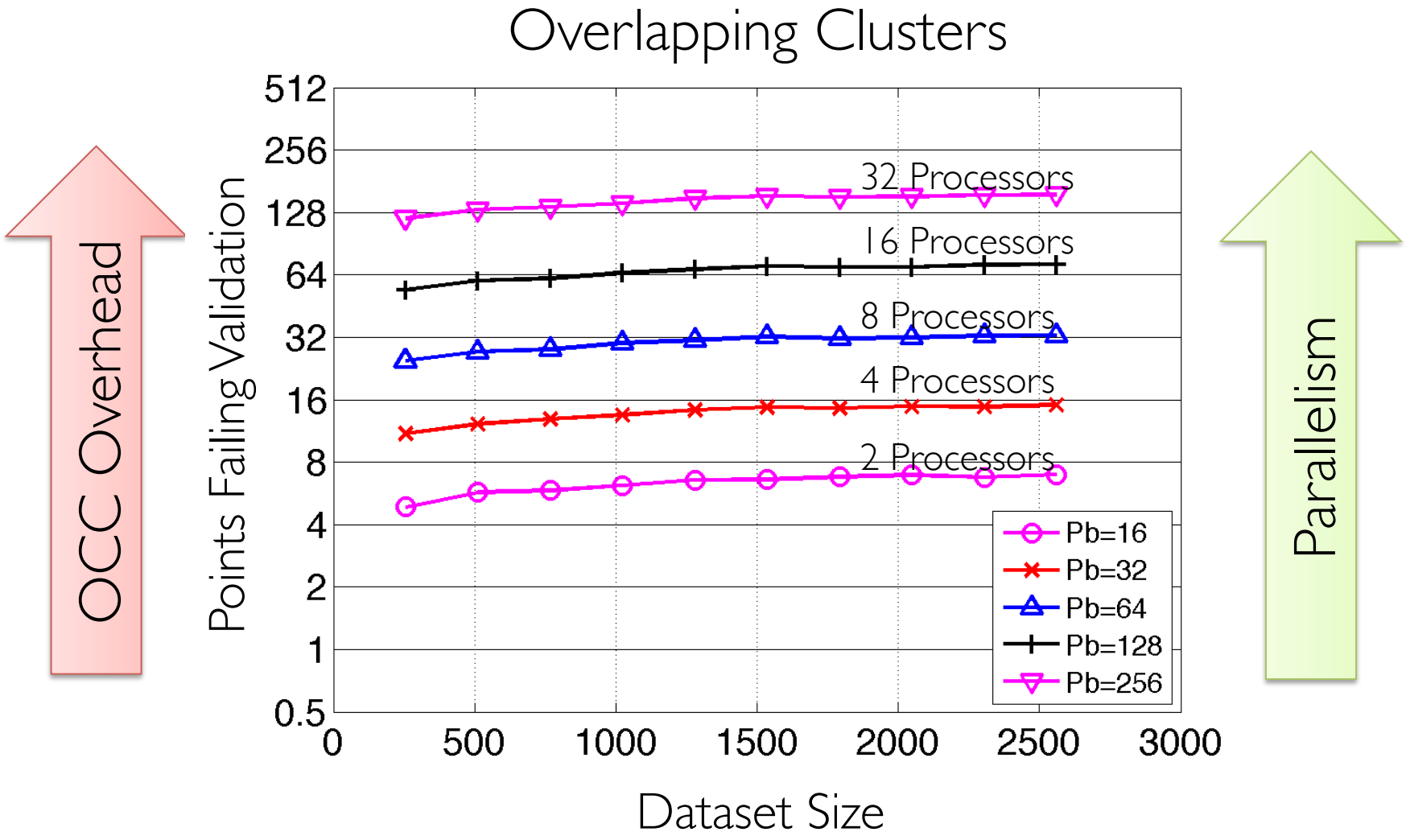
Theorem: Assuming well spaced clusters the expected overhead of OCC DP-means does not depend on data size.

Concurrency

Empirical Validation Failure Rate

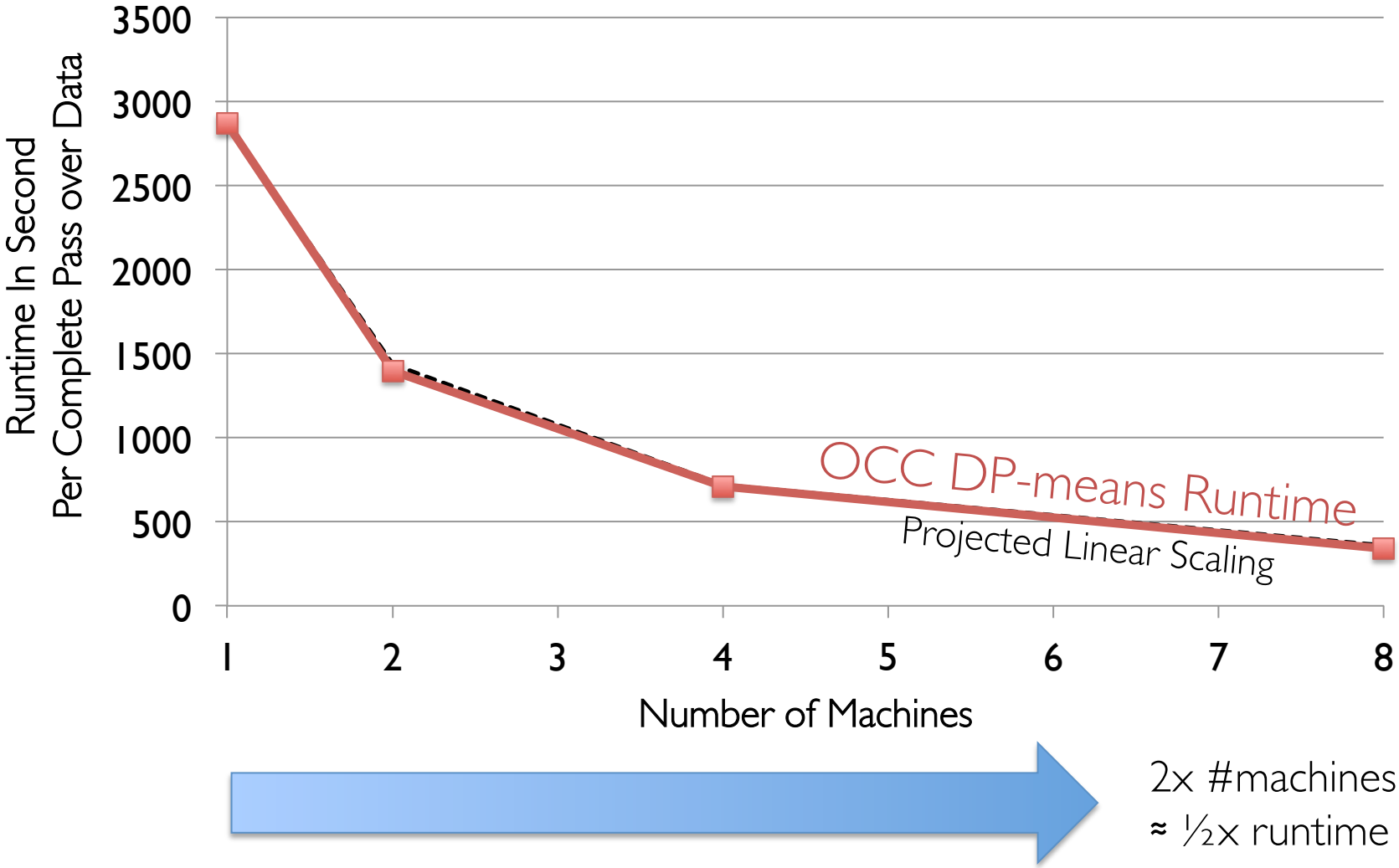


Empirical Validation Failure Rate

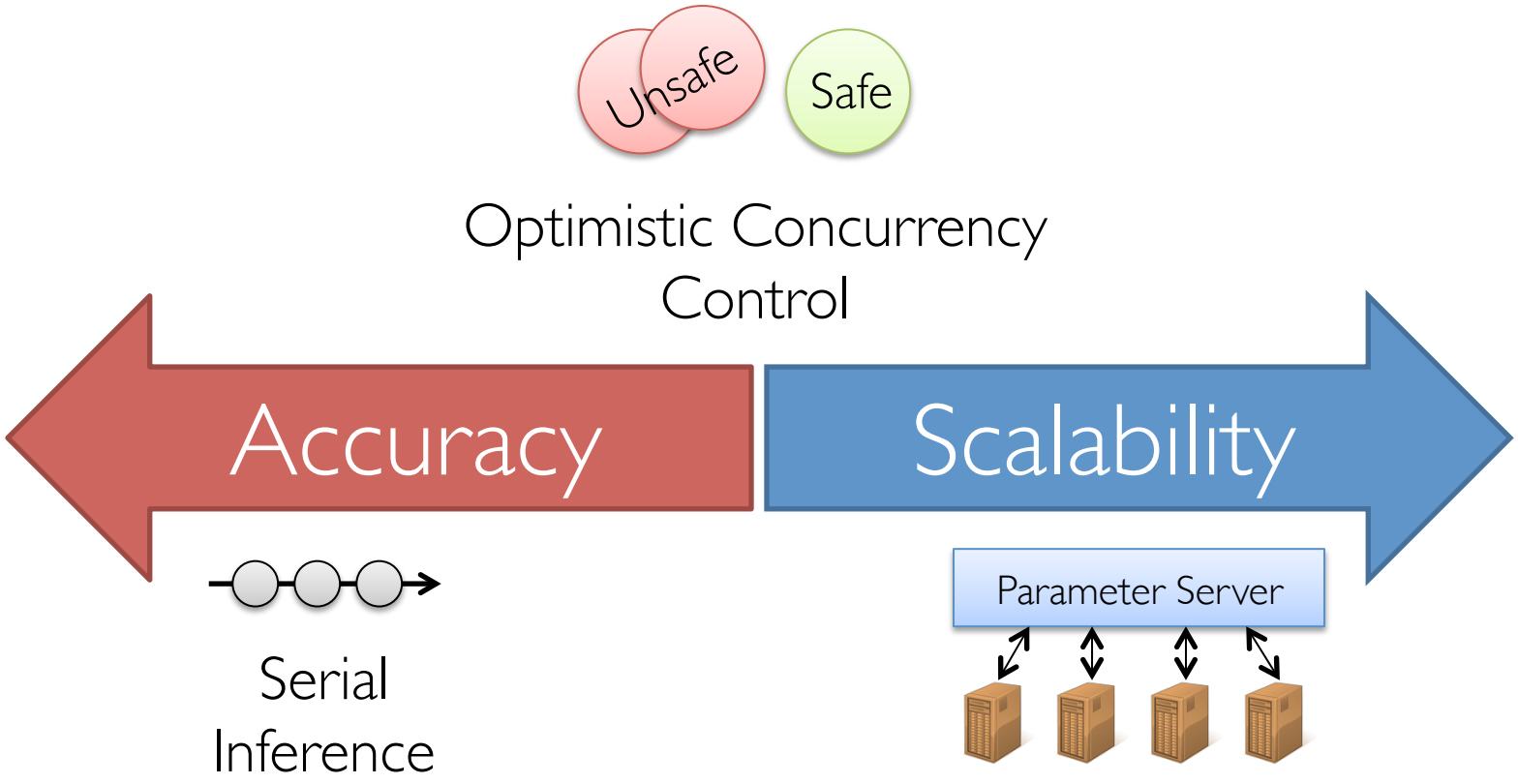


Distributed Evaluation Amazon EC2

~140 million data points; 1, 2, 4, 8 machines



Optimistic Future for Optimistic Concurrency Control



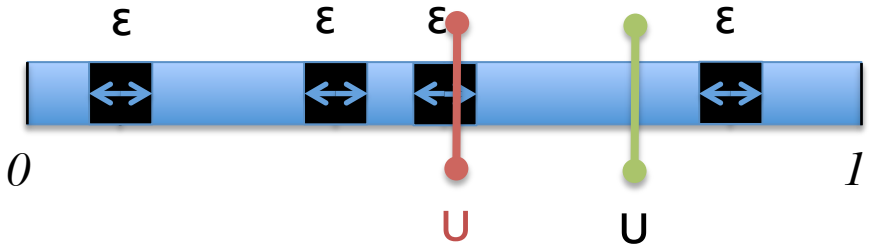
Thank You

Questions?

Joseph E. Gonzalez
jegonzal@eecs.berkeley.edu

OCC Collapsed Gibbs for LDA

Maintain epsilon intervals on the conditional:

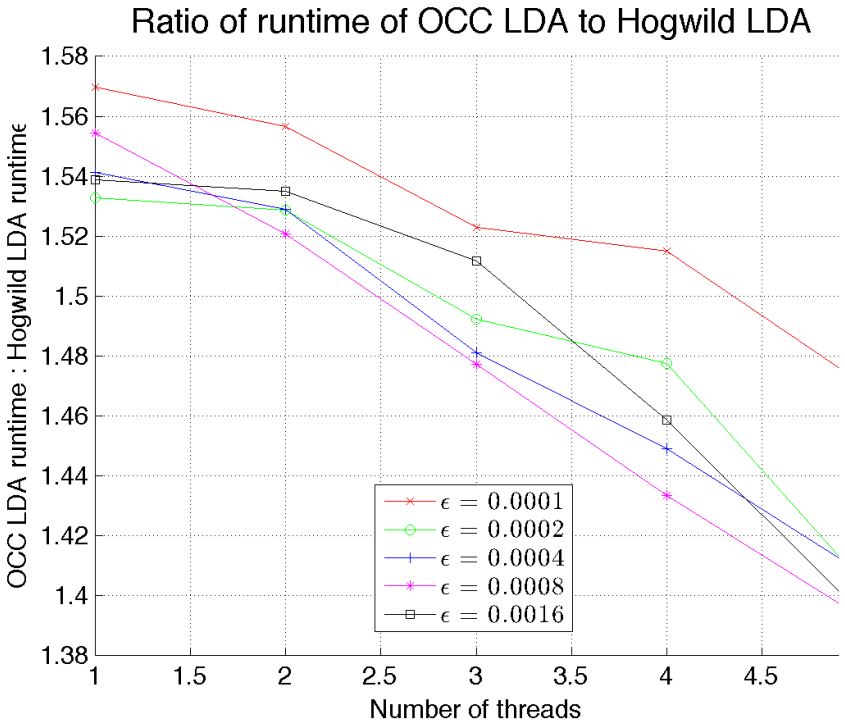


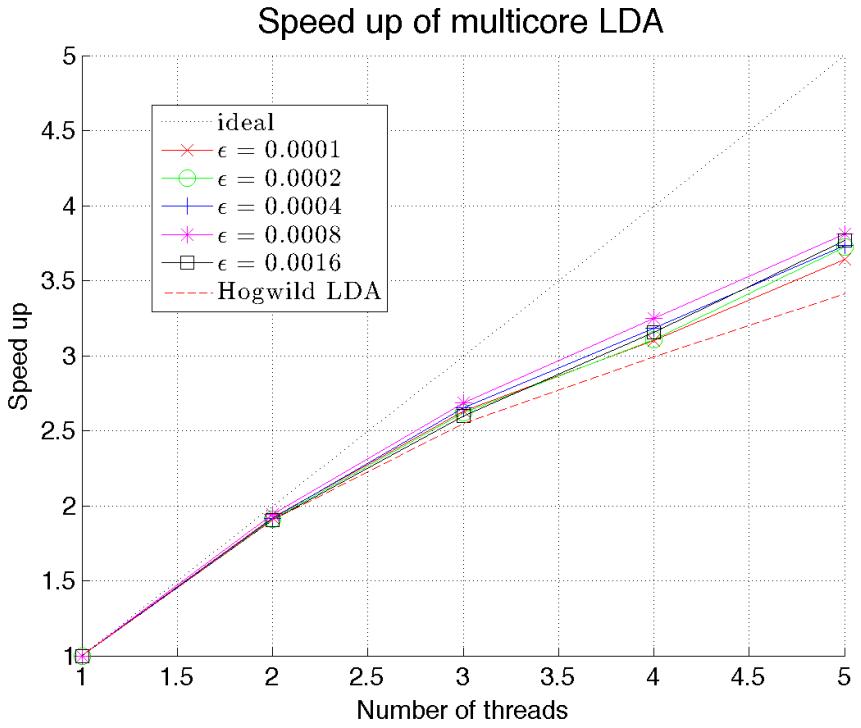
System ensures conditionals are **ε−**accurate

Validate: Accept draws that land outside interval

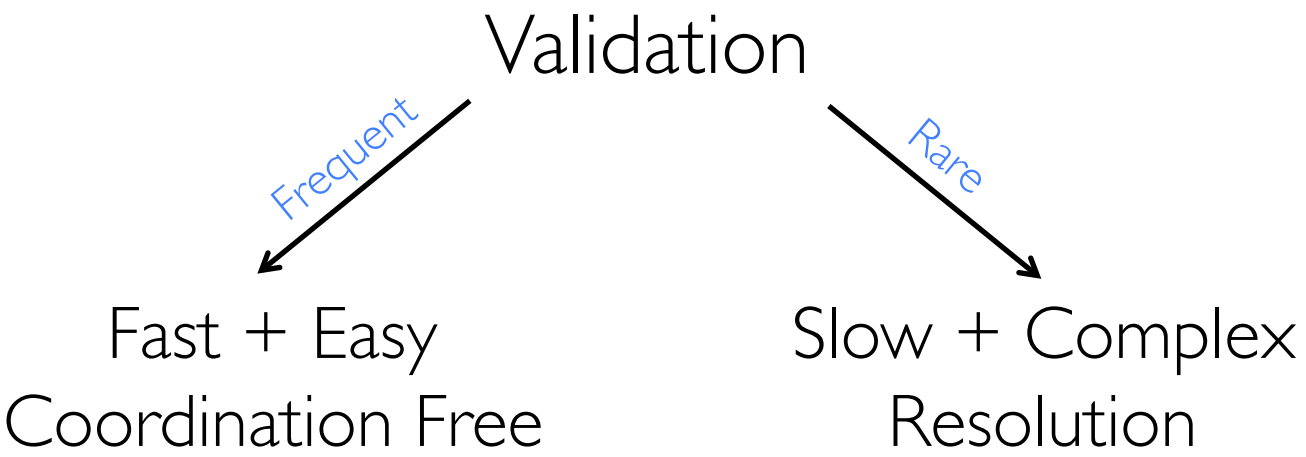
Resolution: Serially resample rejected tokens

OCC Collapsed Gibbs for LDA





Optimism for Optimistic Concurrency Control



Enable decades of work in *serial Bayesian inference* to be extended to the parallel setting.