COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

CentralVR: Efficient Distributed SGD with Variance Reduction

Soham De & Tom Goldstein
University of Maryland

OVERVIEW

What is SGD & why won't it scale?

What is variance reduction?

How can we use VR to boost the scalability of SGD?

MOST MODEL FITTING PROBLEMS LOOK LIKE THIS

$$f(x) = \frac{1}{M} \sum_{i=1}^{M} f(x, d_i)$$

$$\nabla f(x) = \frac{1}{M} \sum_{i=1}^{M} \nabla f(x, d_i)$$

MOST MODEL FITTING PROBLEMS LOOK LIKE THIS

$$f(x) = \frac{1}{M}\sum_{i=1}^{M} f(x, d_i)$$

$$\nabla f(x) = \frac{1}{M}\sum_{i=1}^{M} \nabla f(x, d_i)$$

Applications

SVM

logistic regression

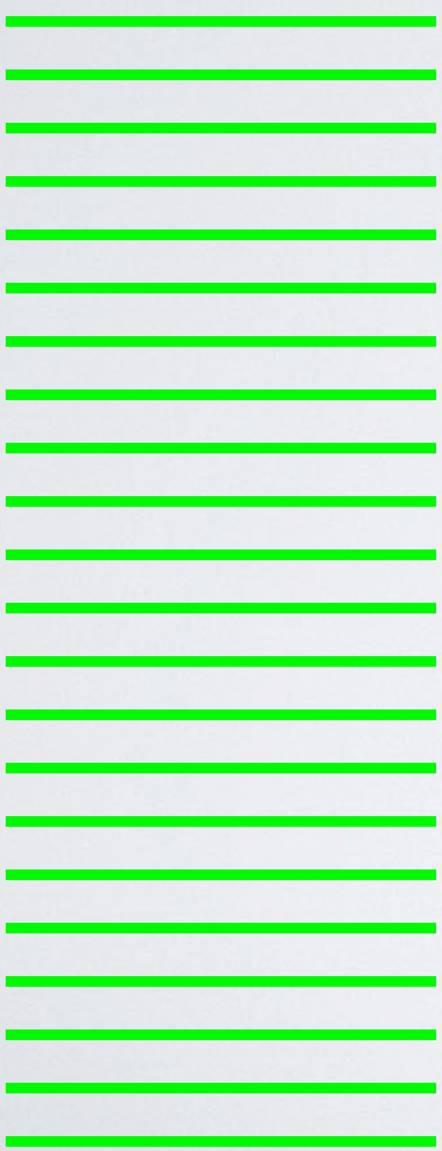
neural nets

matrix factorization

blah blah blah

SGD

select data



compute gradient

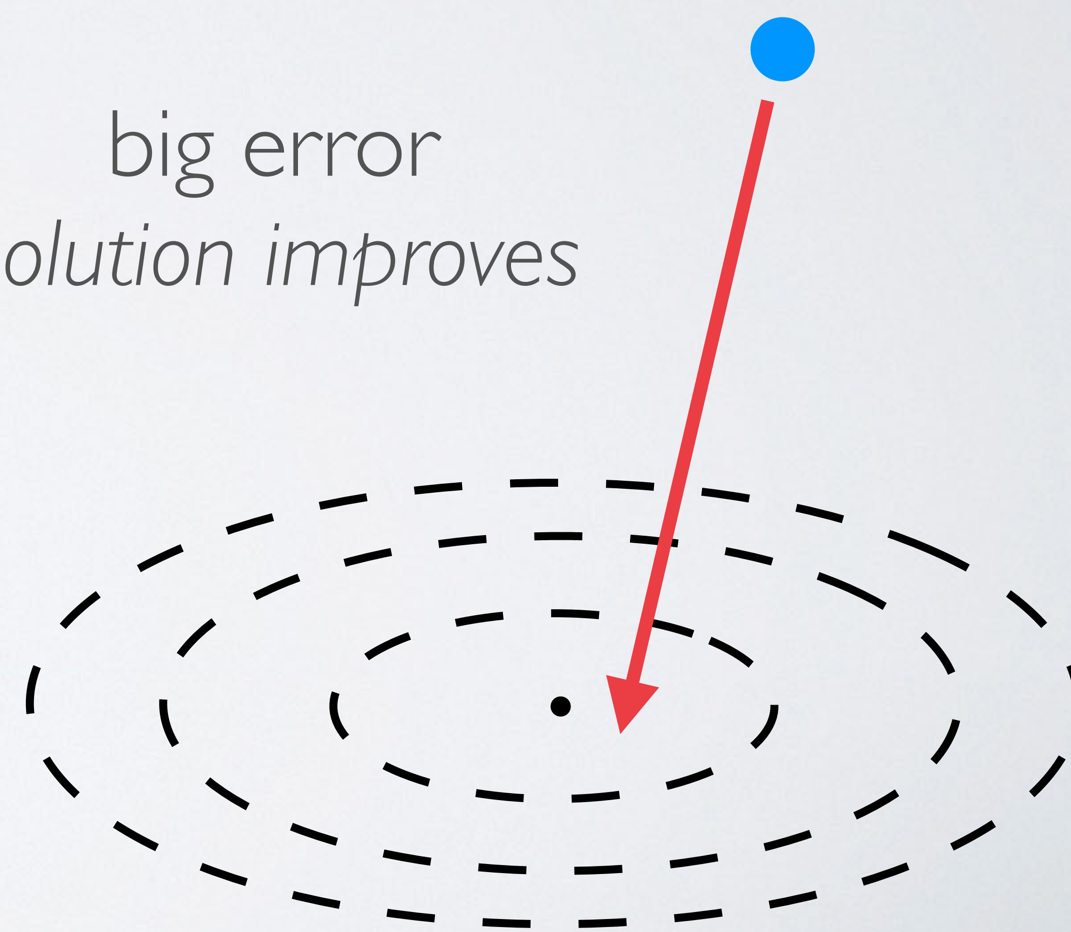
$$g^k = \frac{1}{M} \sum_{i=1}^{M} \nabla f(x, d_i)$$

update

$$x^{k+1} = x^k - \tau_k g^k$$

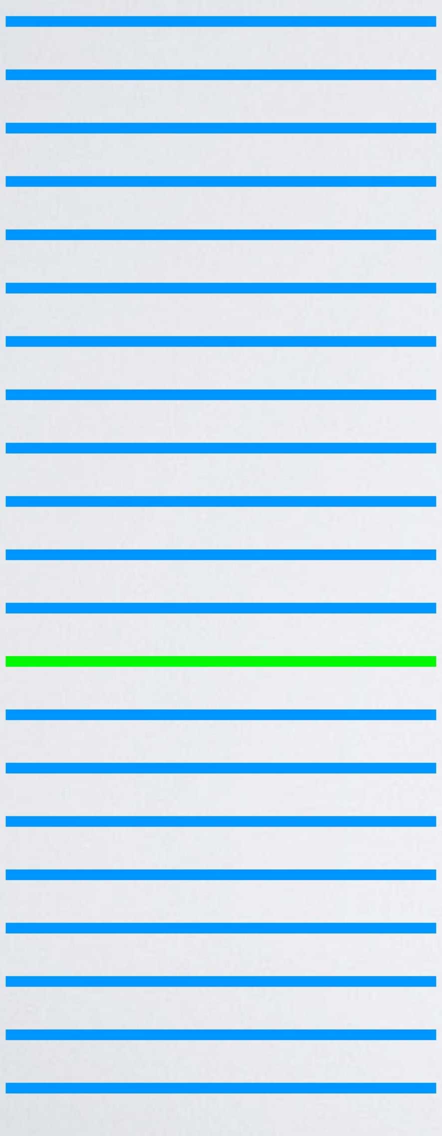
big error

solution improves



SGD

select data



compute gradient

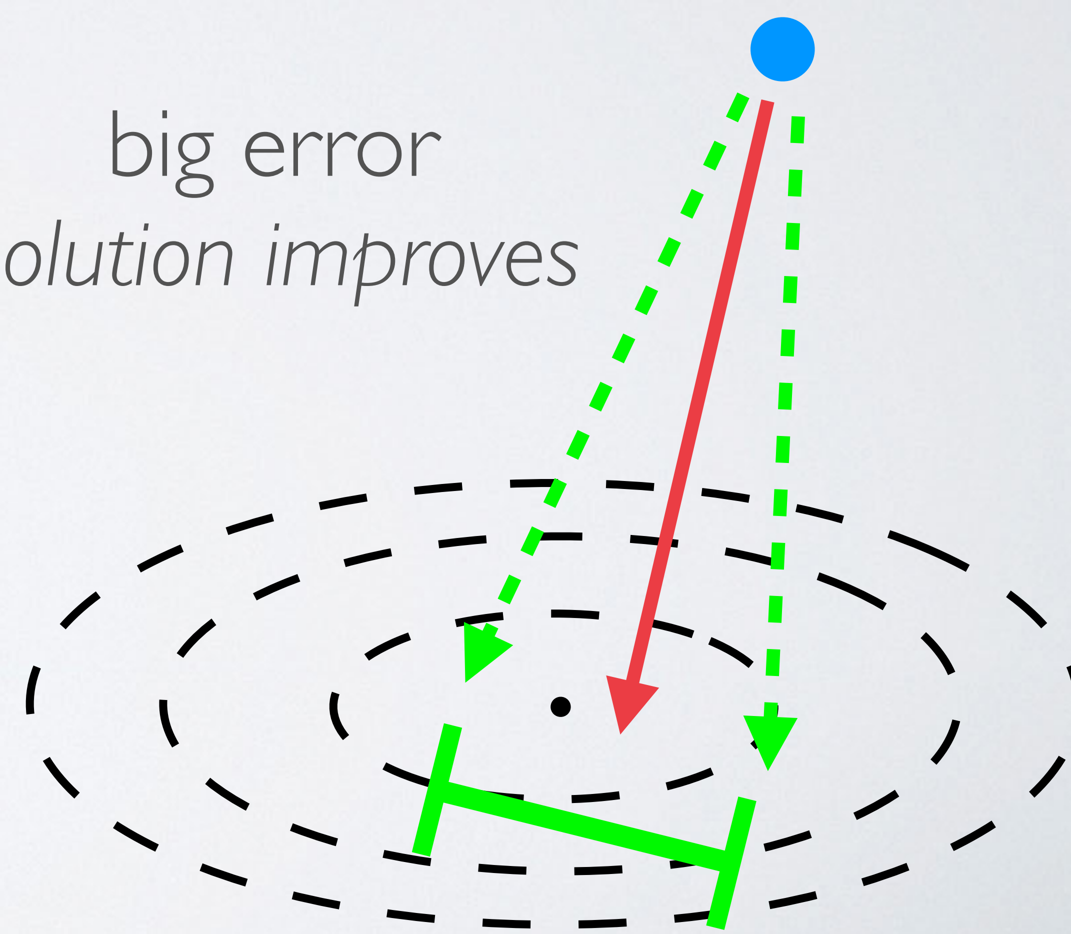
$$g^k \approx \nabla f(x, d_{12})$$

update

$$x^{k+1} = x^k - \tau_k g^k$$

big error

solution improves



SGD

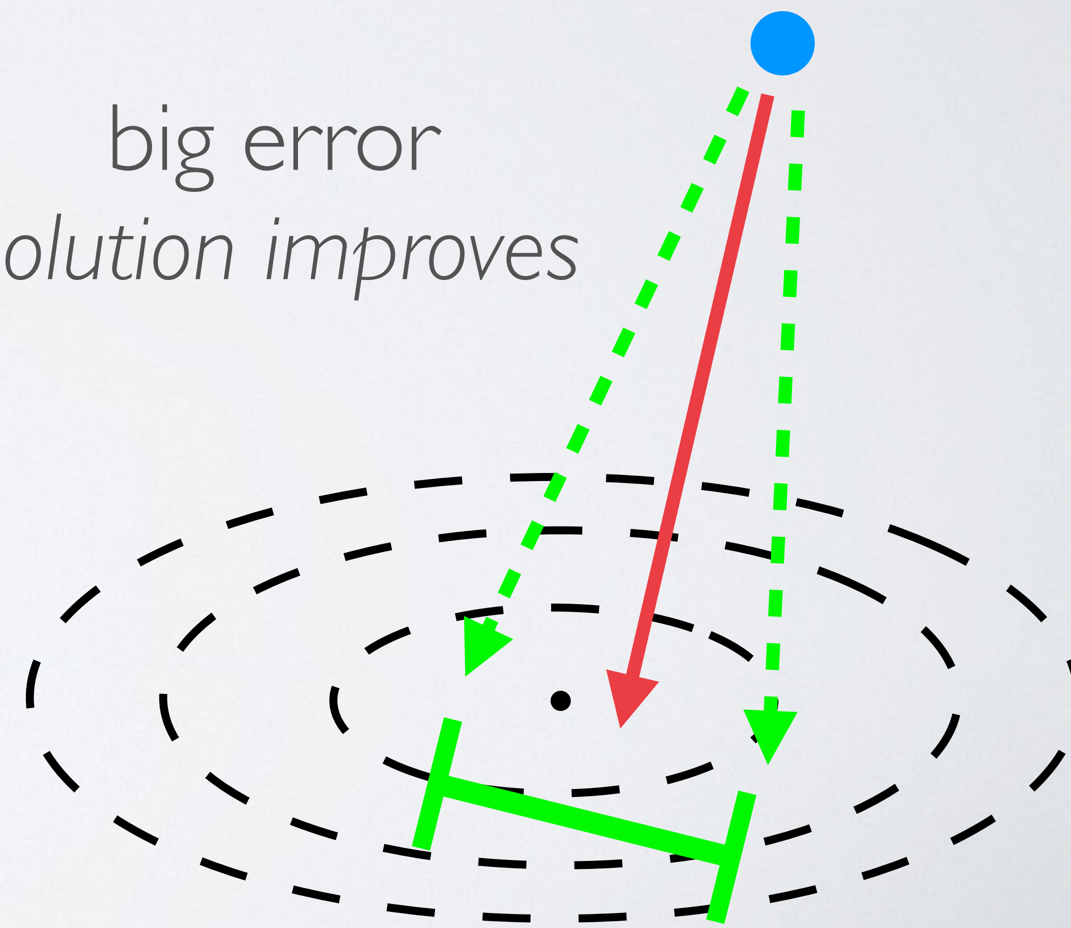
select data

compute gradient

$$g^k \approx \nabla f(x, d_8)$$

update

$$x^{k+1} = x^k - \tau_k g^k$$



big error

solution improves

SGD

select data

compute gradient

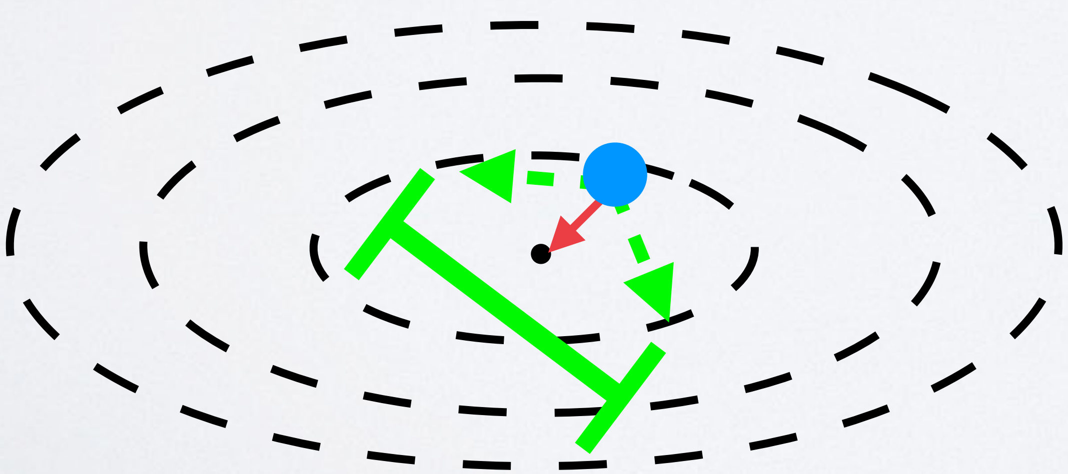
$$g^k \approx \nabla f(x, d_8)$$

update

$$x^{k+1} = x^k - \tau_k g^k$$

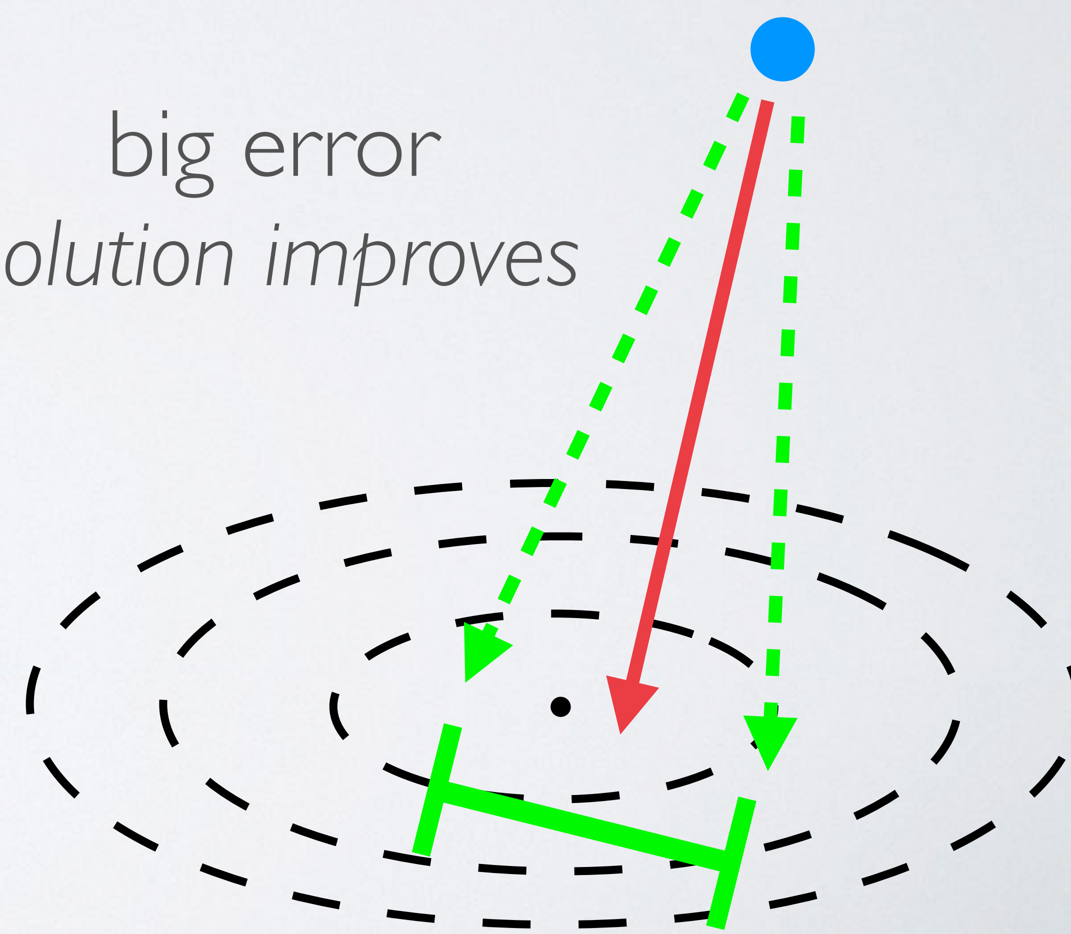
small error

solutions gets worse



big error

solution improves



SGD

select data

compute gradient

$$g^k \approx \nabla f(x, d_8)$$

update

$$x^{k+1} = x^k - \tau_k g^k$$

Error must decrease
as we approach solution

classical solution

shrink stepsize

$$\lim_{k \to \infty} \tau_k = 0$$

slow convergence

$$O(1/\sqrt{k})$$

SGD

select data

compute gradient

$$g^k \approx \nabla f(x, d_8)$$

update

$$x^{k+1} = x^k - \tau_k g^k$$

Error must decrease
as we approach solution

variance reduction solution

make gradient more accurate

preserve fast convergence

SGD

select data

compute gradient

update

$$g^k \approx \nabla f(x, d_8) \; - \; \text{error}^8 \longrightarrow x^{k+1} = x^k - \tau_k g^k$$

Error must decrease
as we approach solution

variance reduction solution

make gradient more accurate

preserve fast convergence

VR APPROACHES

SAGA
Defazio, Bach, Lacoste-Julian, 2014

SAG
Le Roux, Schmidt, Bach, 2013

SVRG
Johnson, Zhang, 2013

many more…

Central VR
A VR approach targeting **distributed** ML

Also, we propose *distributed* variants of these methods

"Efficient Distributed SGD with Variance Reduction," ICDM 2016

CENTRAL VR

gradient tableau First epoch

CENTRAL VR

gradient tableau

First epoch

$\nabla f_1(x_m^1)$
$\nabla f_2(x_m^2)$
$\nabla f_3(x_m^3)$
$\vdots$
$\nabla f_{n-1}(x_m^{n-1})$
$\nabla f_n(x_m^n)$

CENTRAL VR

gradient tableau

$\nabla f_1(x_m^1)$
$\nabla f_2(x_m^2)$
$\nabla f_3(x_m^3)$
$\vdots$
$\nabla f_{n-1}(x_m^{n-1})$
$\nabla f_n(x_m^n)$

Approximate true gradient over last epoch

$$\bar{g}_m = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_m^i)$$

CENTRAL VR

gradient tableau

$\nabla f_1(x_m^1)$
$\nabla f_2(x_m^2)$
$\nabla f_3(x_m^3)$
$\vdots$
$\nabla f_{n-1}(x_m^{n-1})$
$\nabla f_n(x_m^n)$

Approximate true gradient over last epoch

$$\bar{g}_m = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_m^i)$$

corrected gradient

$$\nabla f_3(x_{m+1}^3) \; - \; \underbrace{\left(\nabla f_3(x_m^3) - \bar{g}_m\right)}_{\text{error}}$$

new gradient

CENTRAL VR

gradient tableau

$\nabla f_1(x_m^1)$
$\nabla f_2(x_m^2)$
$\nabla f_3(x_{m+1}^3)$
$\vdots$
$\nabla f_{n-1}(x_m^{n-1})$
$\nabla f_n(x_m^n)$

Approximate true gradient over last epoch

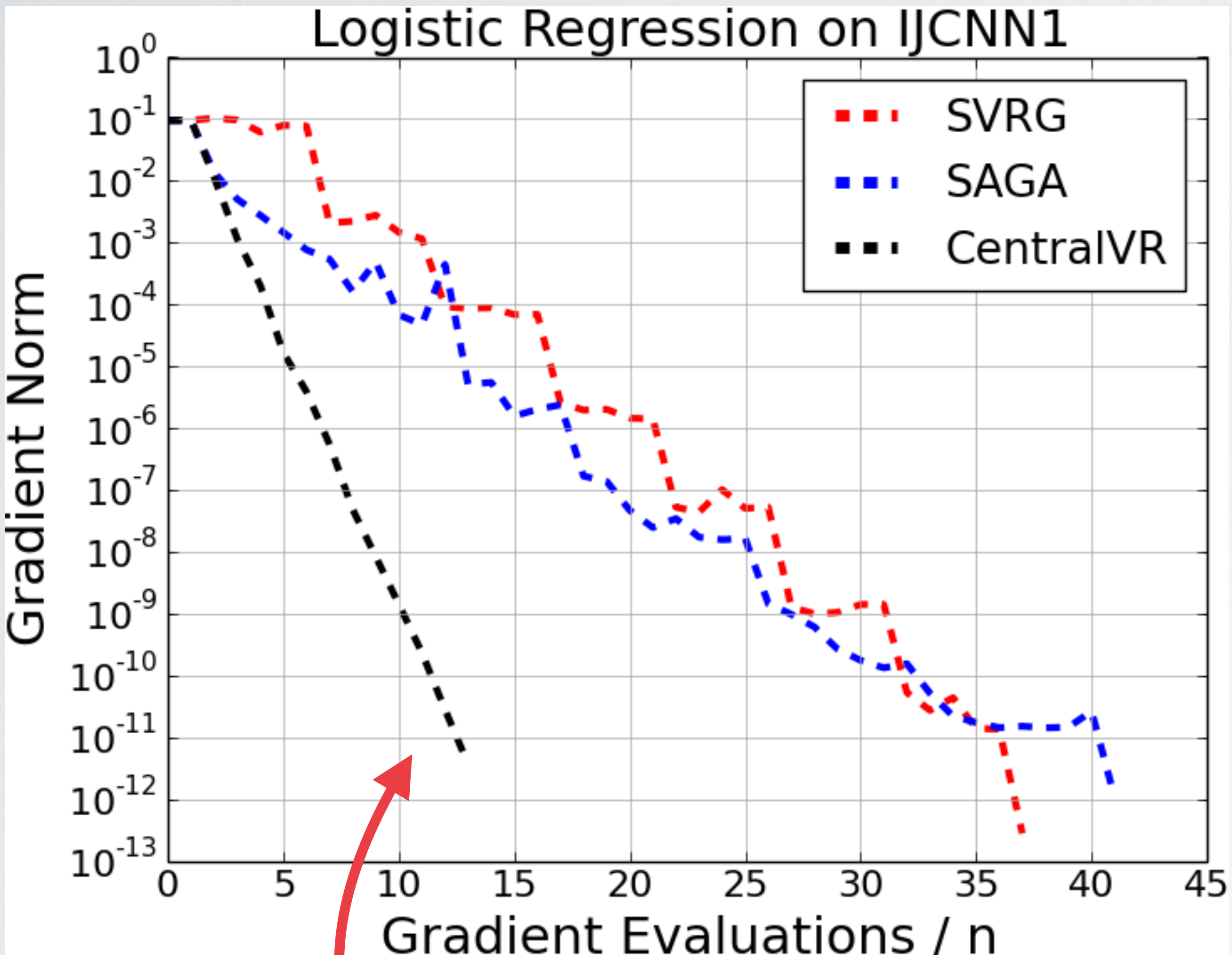
$$\bar{g}_m = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_m^i)$$

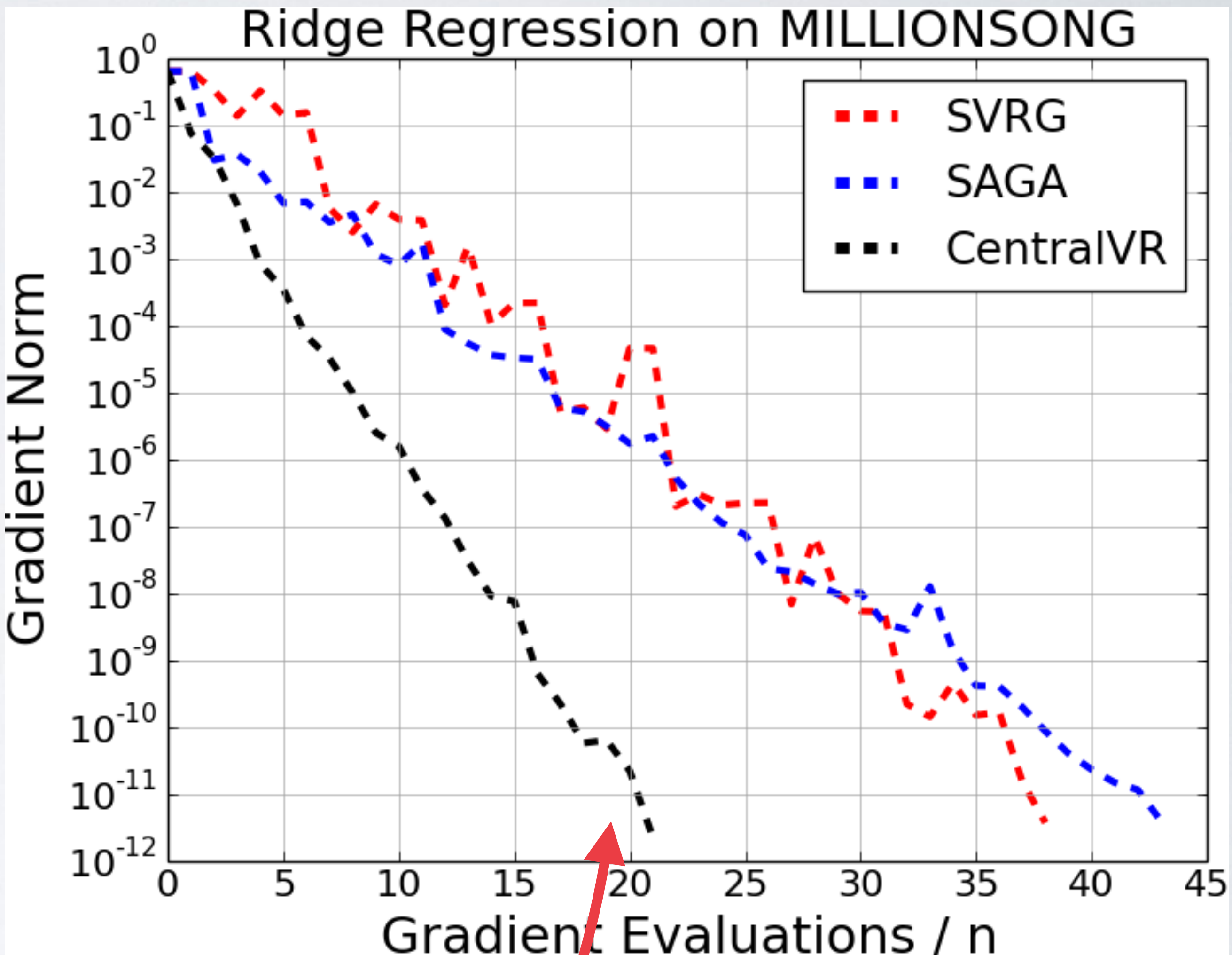
corrected gradient

$$\nabla f_3(x_{m+1}^3) - \underbrace{(\nabla f_3(x_m^3) - \bar{g}_m)}_{\text{error}}$$

new gradient

SINGLE-WORKER RESULTS





Roughly 2X speedup over other methods

WHAT'S WRONG WITH DISTRIBUTED SGD

diverging paths

$$x^{k+1} = x^k - \tau_k g^k$$

slow decay of noise

$$g^k = \nabla f(x^k) + \mathrm{noise}^k$$

$$O(1/\sqrt{\mathrm{workers}})$$

WHAT'S WRONG WITH DISTRIBUTED SGD

diverging paths

$$x^{k+1} = x^k - \tau_k g^k$$

Use GLOBAL error corrections to keep workers on same path

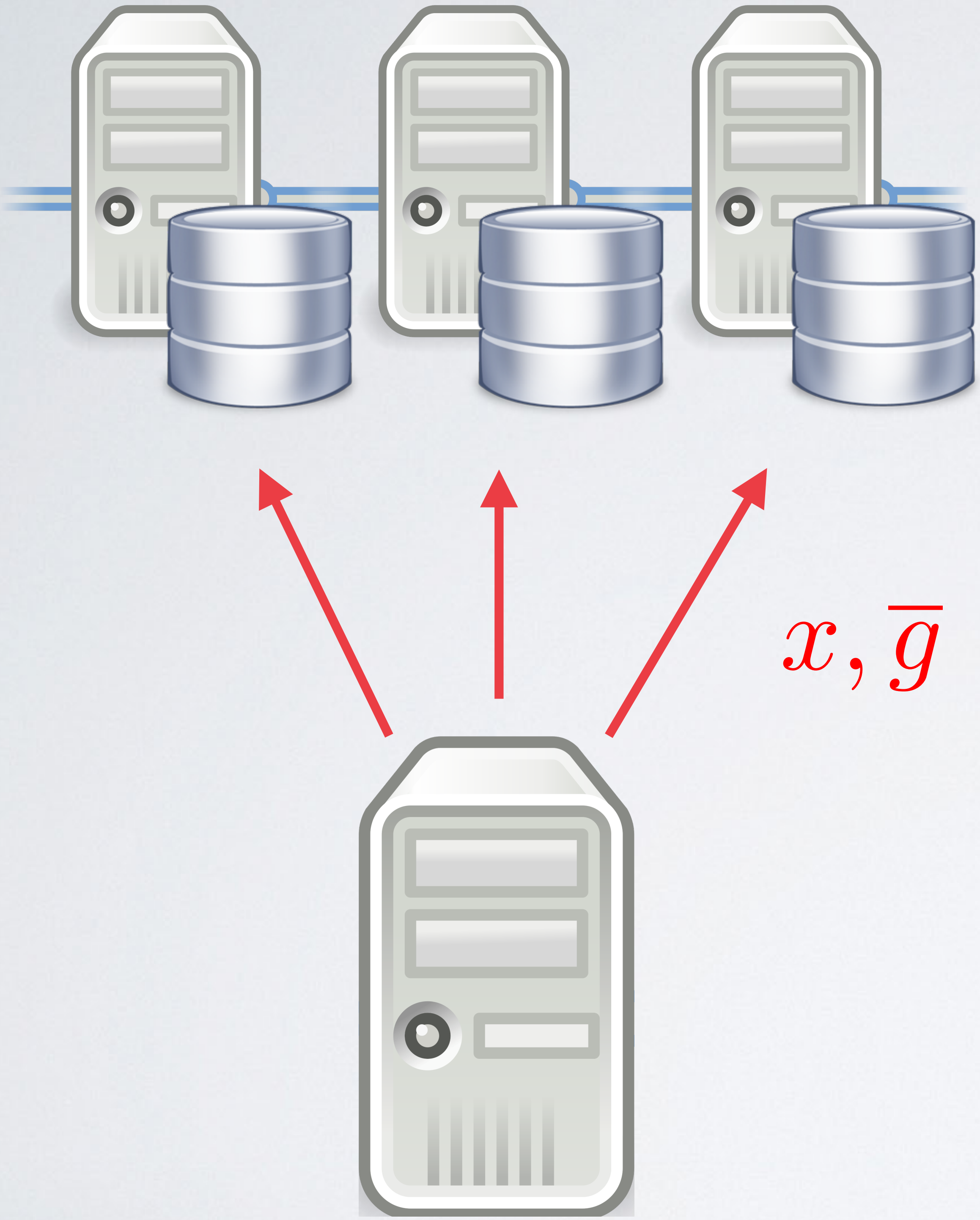
slow decay of noise

$$g^k = \nabla f(x^k) + \text{noise}^k$$

$$O(1/\sqrt{\text{workers}})$$

Use VR methods to reduce error faster than averaging

SYNCHRONOUS CENTRALVR

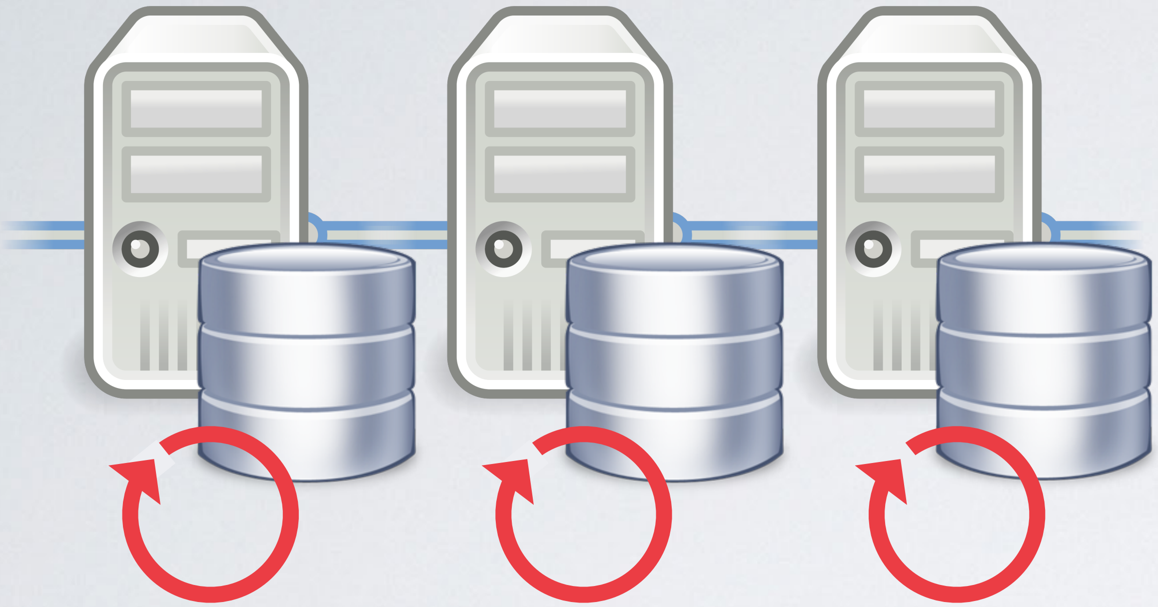


- Each local node maintains **local tableau** of stored gradients
- Local nodes receive current iterate and average gradient from central server

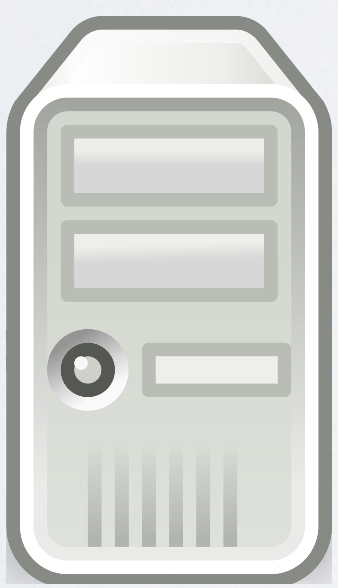
$\bar{g}$ is now **global** average

x is **shared** at start of epoch

SYNCHRONOUS CENTRALVR



one epoch

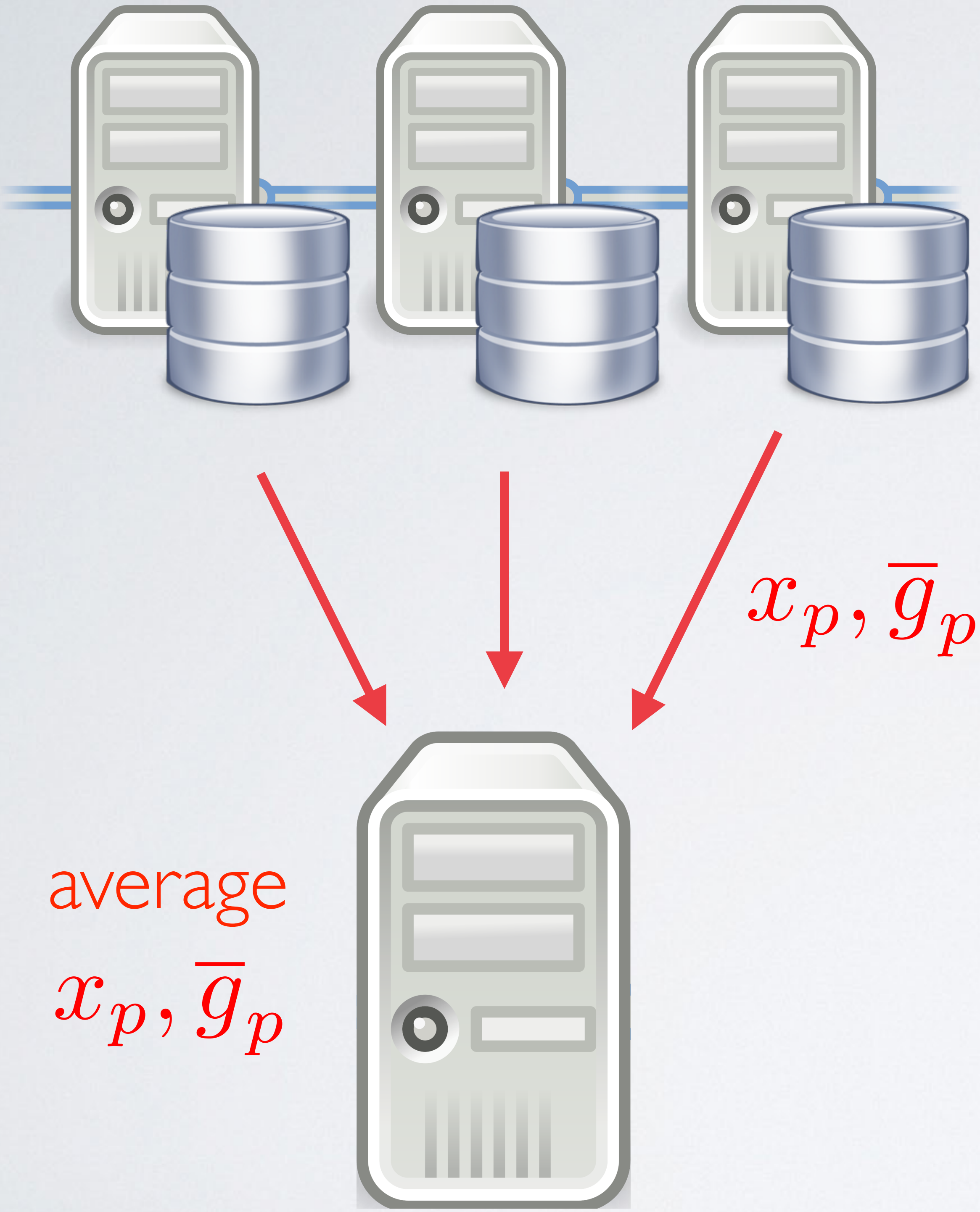


- Each local node maintains **local tableau** of stored gradients
- Local nodes receive current iterate and average gradient from central server
- Each local node runs one epoch of CentralVR

$$\nabla f_3(x^3_{m+1}) \;-\; \underbrace{(\nabla f_3(x^3_m) - \bar{g}_m)}_{\text{error relative to } \textbf{global} \text{ average}}$$

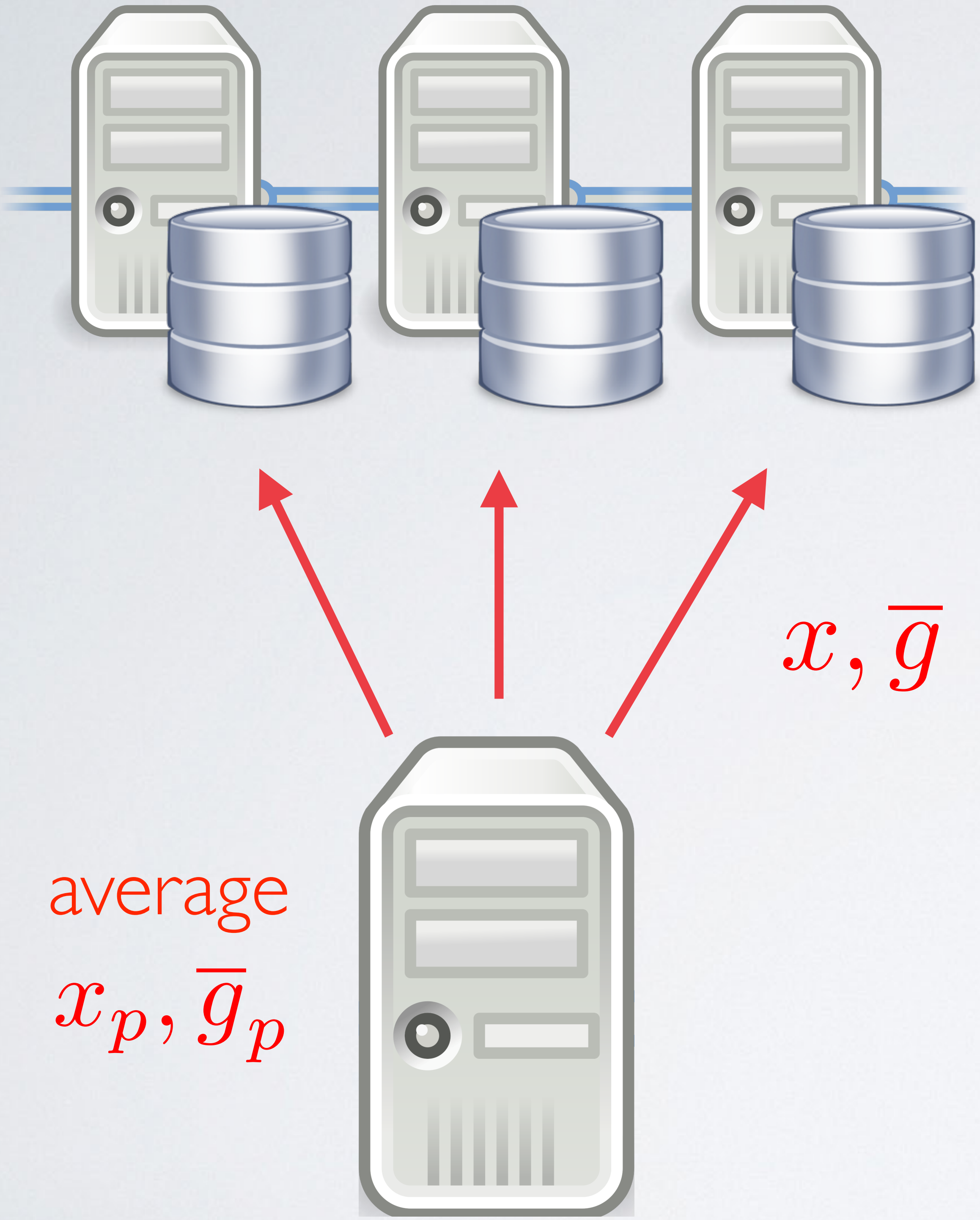
local gradient

SYNCHRONOUS CENTRALVR



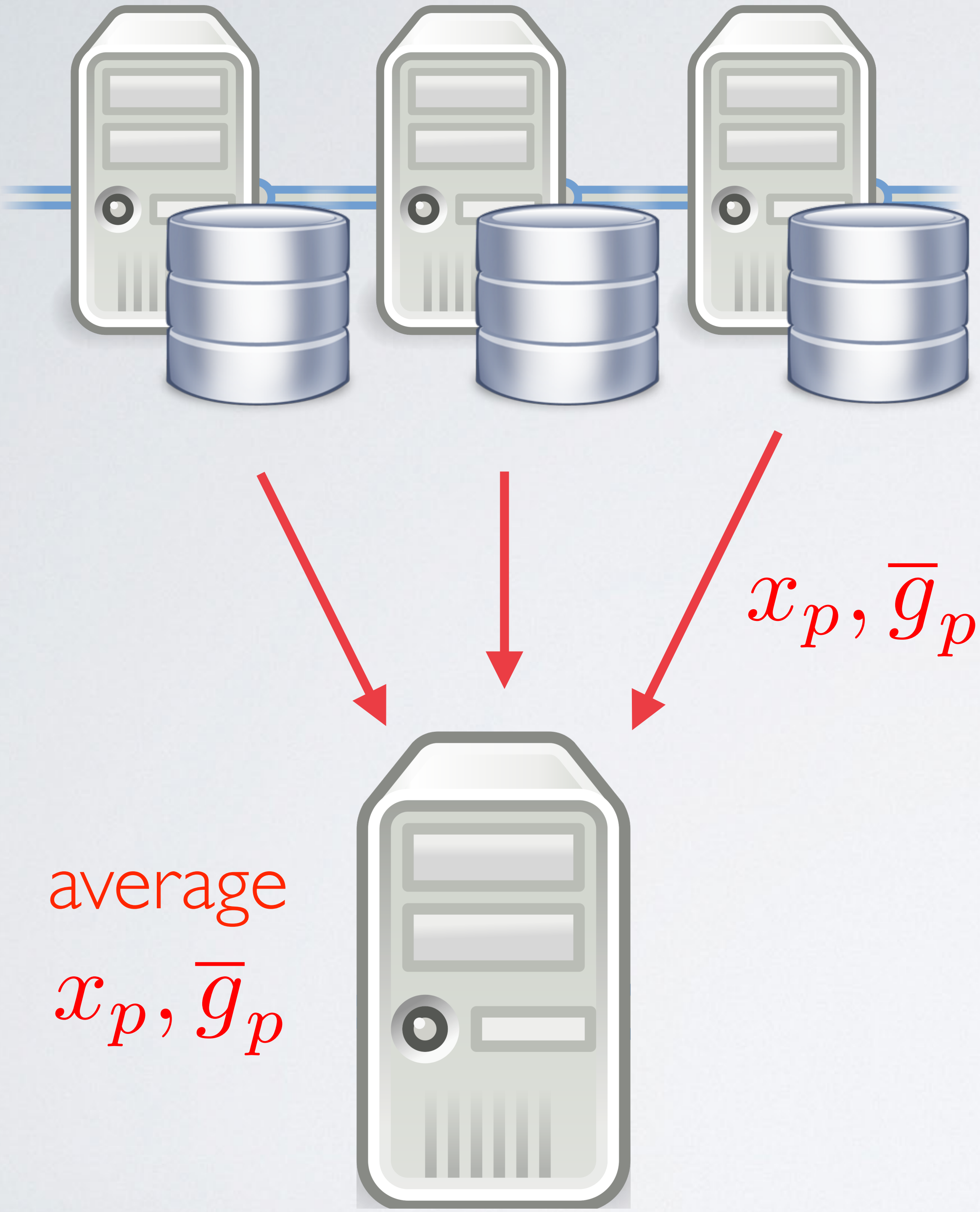
- Each local node maintains **local tableau** of stored gradients
- Local nodes receive current iterate and average gradient from central server
- Each local node runs one epoch of CentralVR
- Send current local iterate and local average gradient

SYNCHRONOUS CENTRALVR



- Each local node maintains **local tableau** of stored gradients
- Local nodes receive current iterate and average gradient from central server
- Each local node runs one epoch of CentralVR
- Send current local iterate and local average gradient
- Central server averages and broadcasts

ASYNCHRONOUS VERSION



Key Difference:
Local node sends back
change in variables

$$\Delta x_p^m = x_p^m - x_p^{m-1}$$

$$\Delta \overline{g}_p^m = \overline{g}_p^m - \overline{g}_p^{m-1}$$

robust to different node speeds

EMPIRICAL RESULTS

Model: Ridge Regression

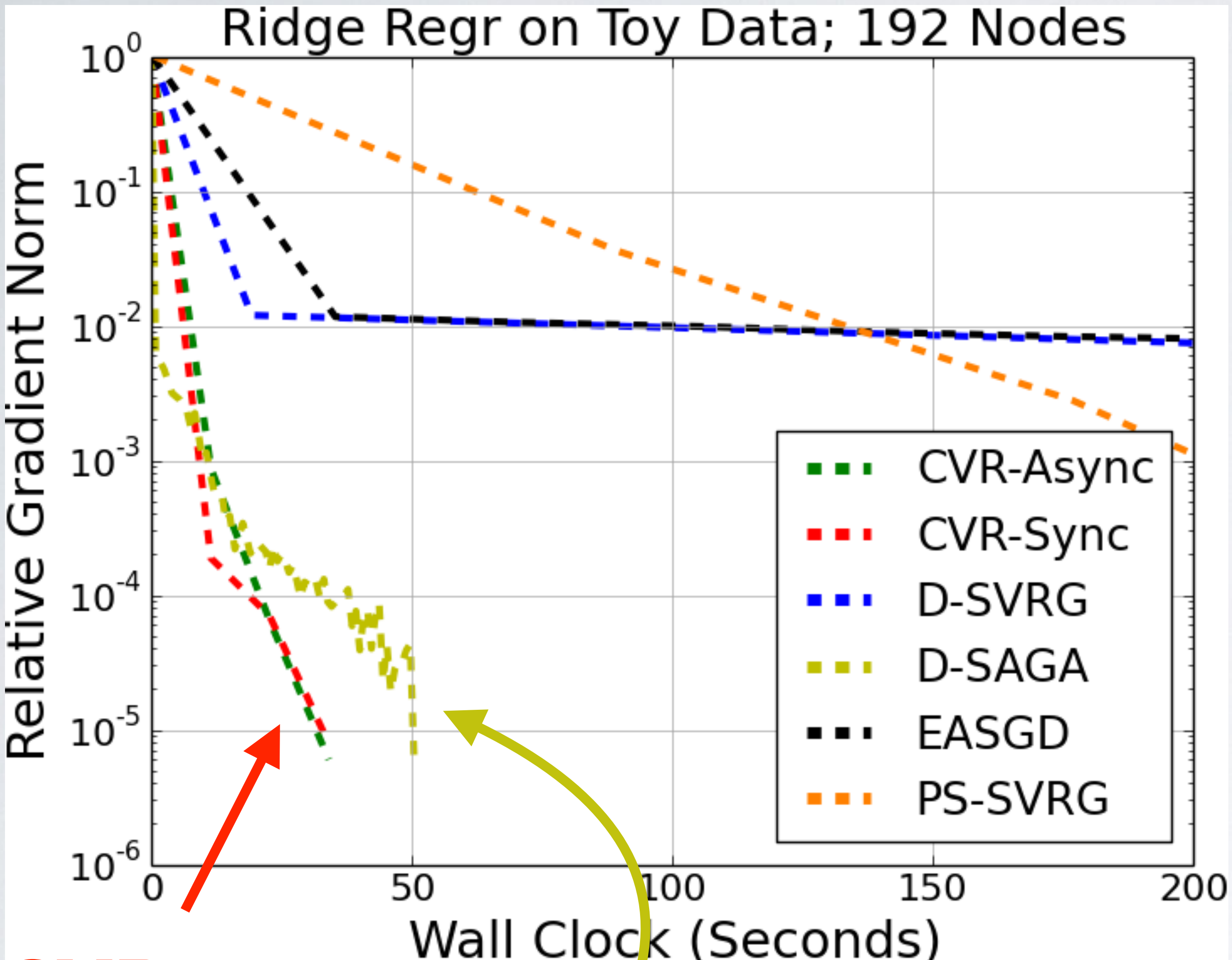
Datasets:
MILLIONSONG for regression: 463,715 samples
Toy data (random A, $b = Ax + \epsilon$): 5000 samples/node

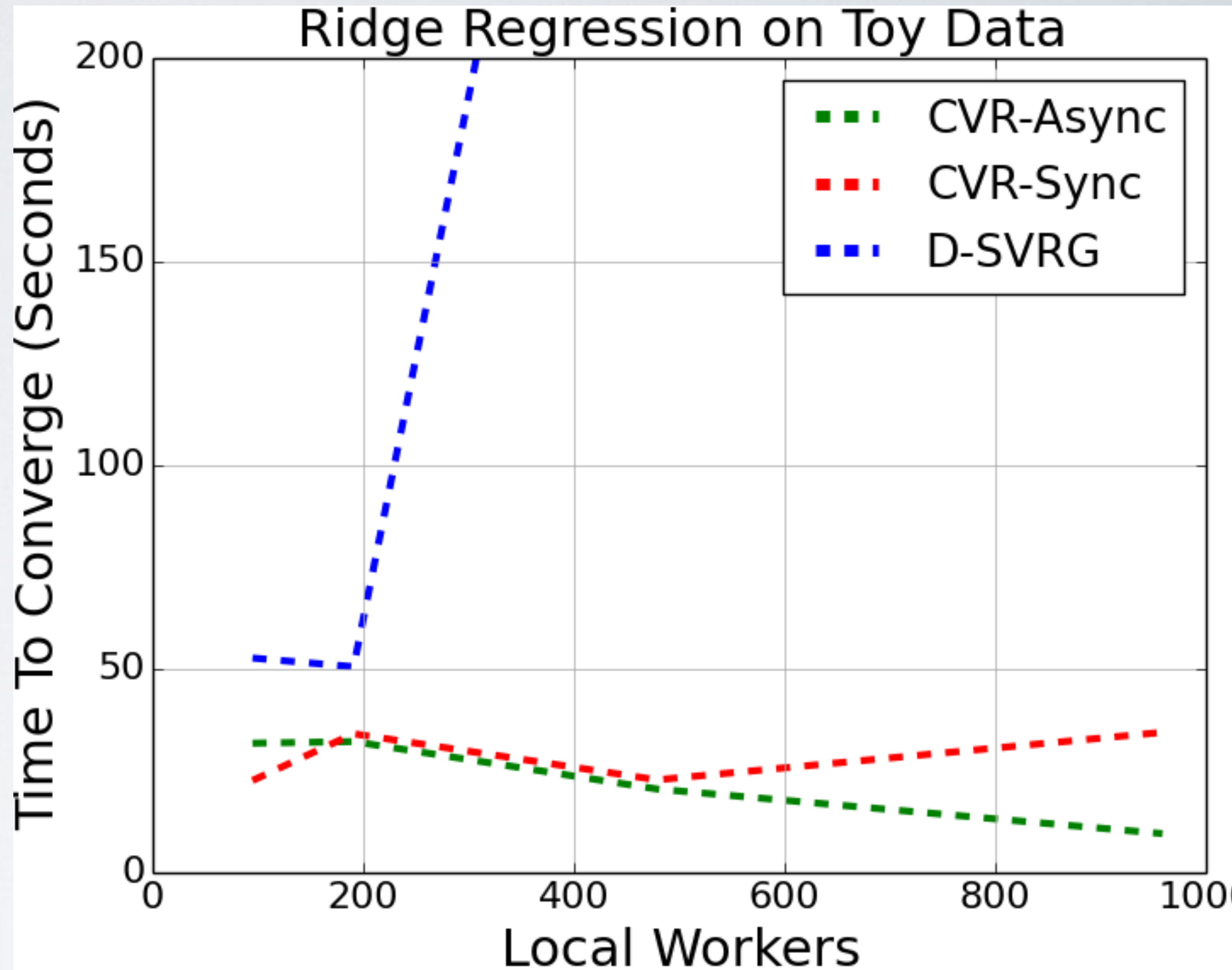
Compared with:

- EASGD (Zhang, Choromanska, Lecun, 15)
- Asynchronous SVRG (Reddi et al, 15)
- Distributed SAGA (in CentralVR paper)
- Distributed SVRG (in CentralVR paper)

Check paper for additional experiments

EMPIRICAL RESULTS





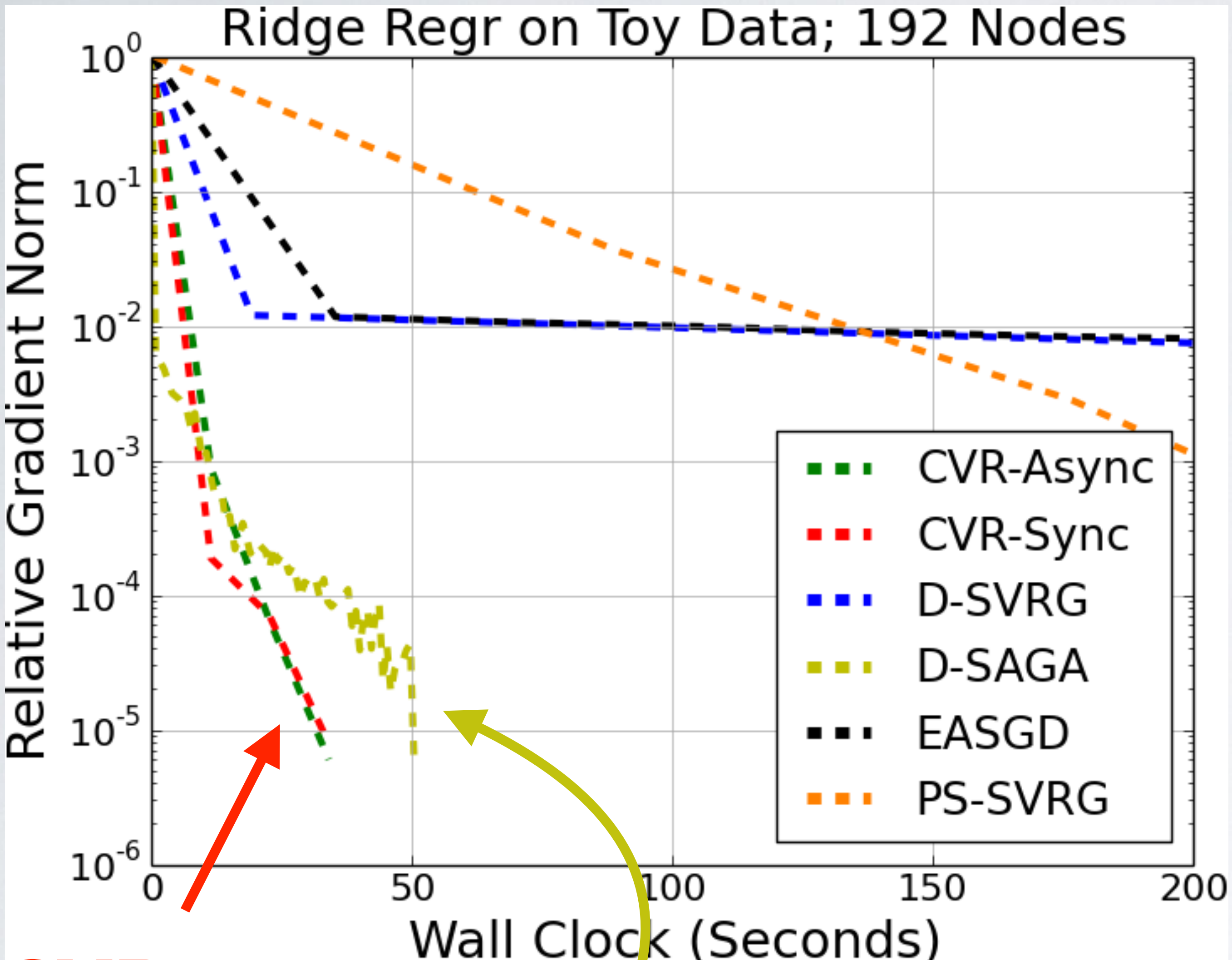
CVR-sync

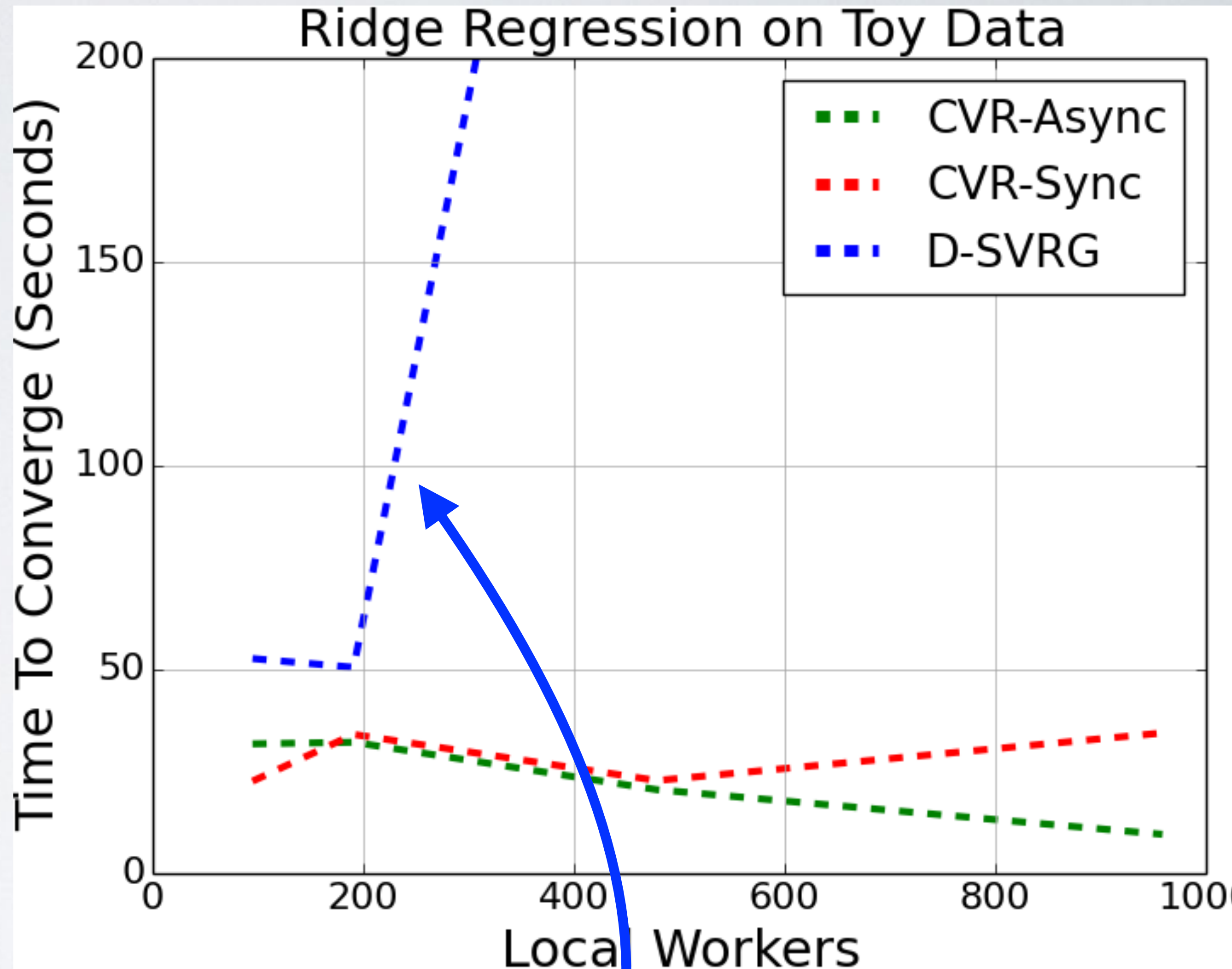
CVR-async CVR-async

Toy data set size increases linearly with number of workers

Maximum toy data set size: 5000*960 = 4,800,000

EMPIRICAL RESULTS





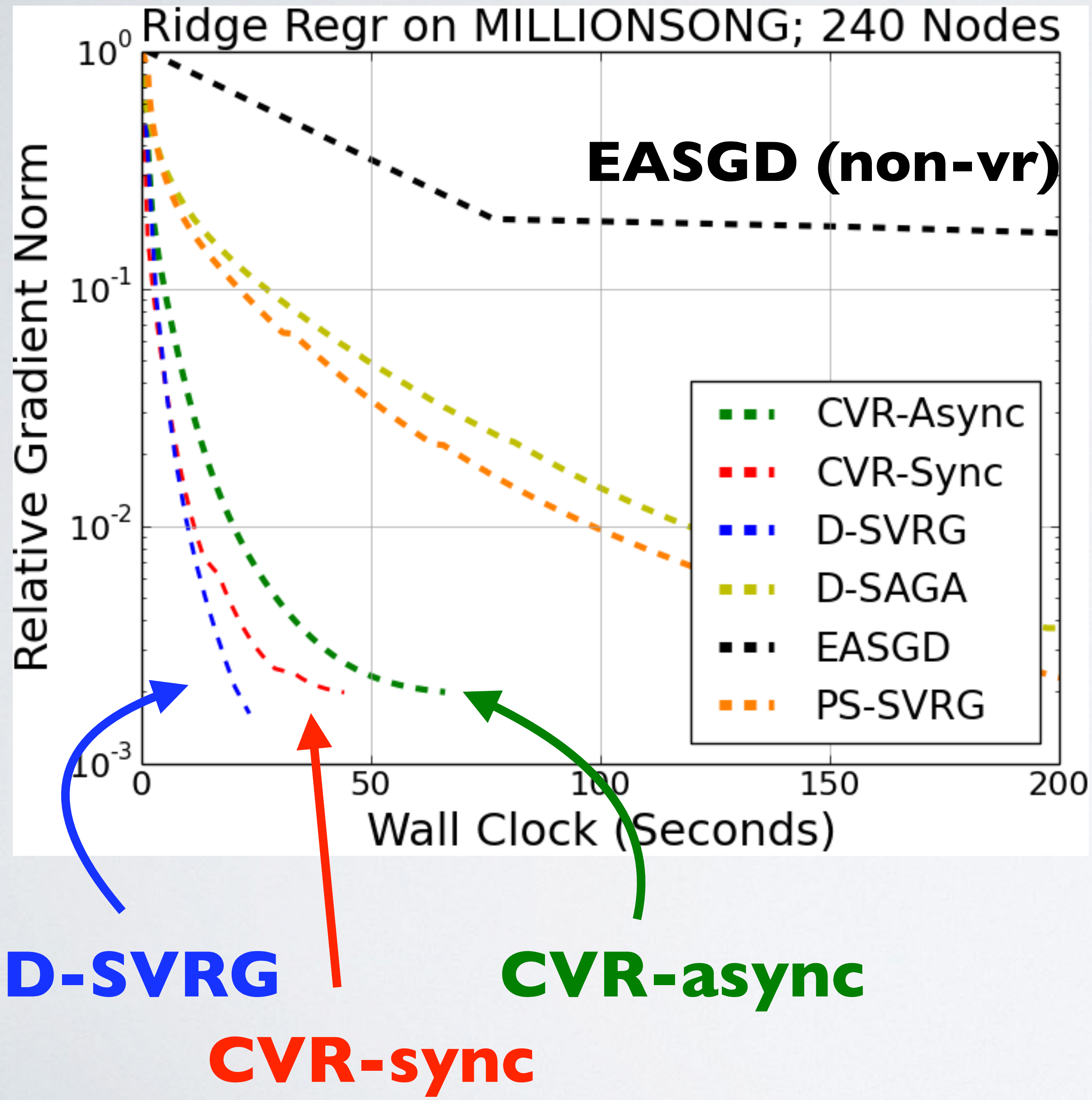
CVR-sync

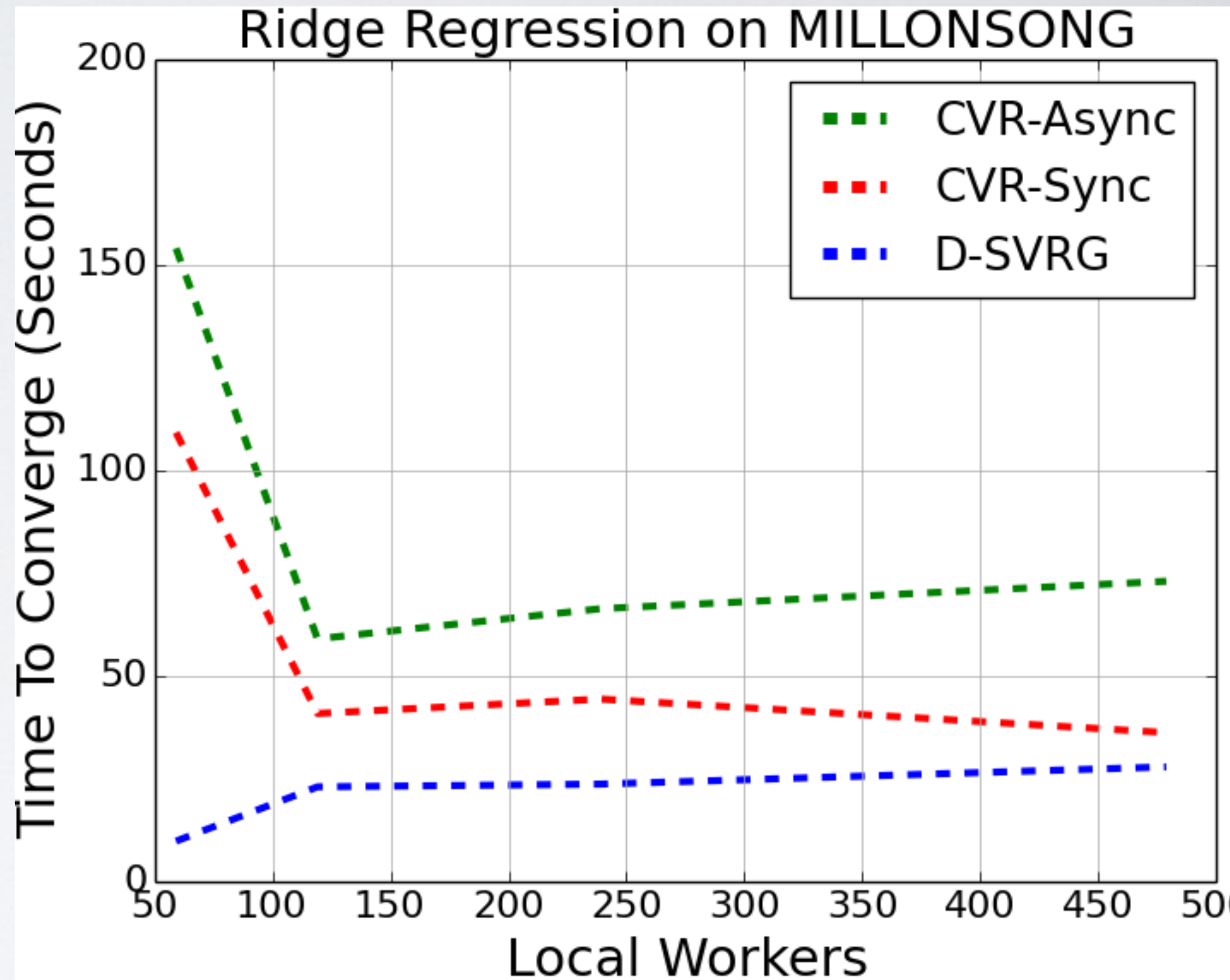
CVR-async **CVR-async**

SCOPE: Scalable Composite Optimization for Learning on Spark, Zhao 2017

Toy data set size increases linearly with number of workers
Maximum toy data set size: 5000*960 = 4,800,000

EMPIRICAL RESULTS





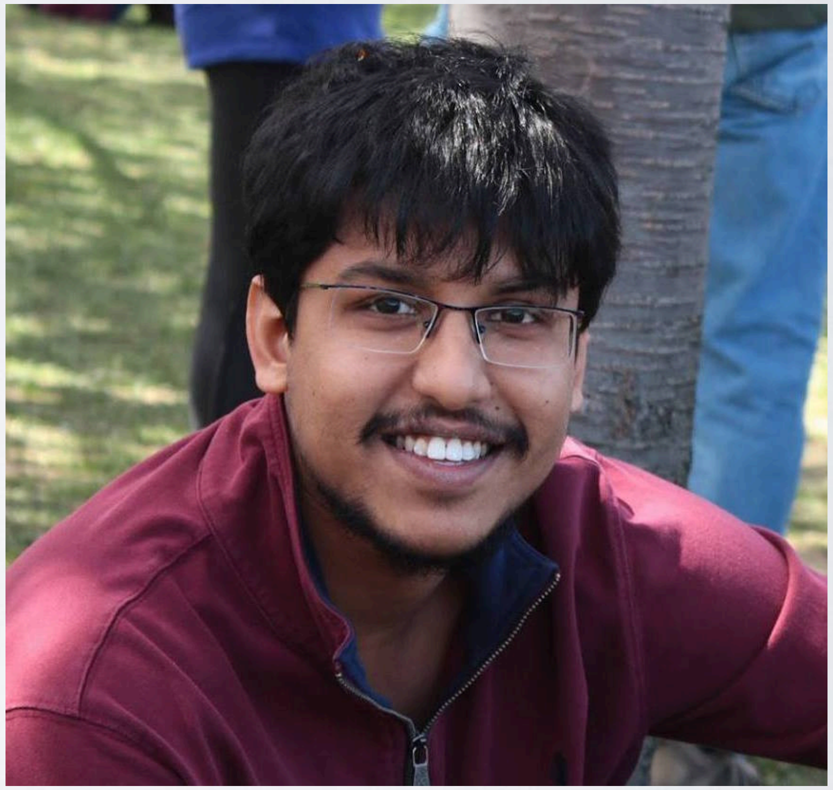
TAKEAWAYS

Distributed Variance Reduction

- Boosts the scalability of SGD to **hundreds** of distributed computing nodes
- **Low** communication costs suitable for large-scale heterogenous distributed environments

THANKS!

Soham De



Feel free to get in touch!

email

sohamde@cs.umd.edu
tomg@cs.umd.edu

Website: https://cs.umd.edu/~sohamde/