Information Theory Workshop, 2018

C^3LES : Codes for Coded Computation that Leverage Stragglers

Anindya Bijoy Das
Li Tang
Aditya Ramamoorthy

November 28, 2018

Electrical and Computer Engineering
Iowa State University

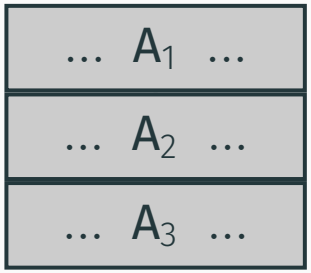
Distributed Matrix-Vector Computation

$$\begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & \dots & a_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_m \end{bmatrix}$$

$\mathbf{A}$ $\quad$ x

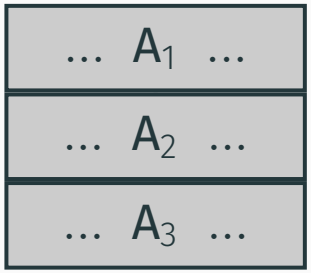
Matrix **A** is typically too big to use just one node for computation.

Distributed Matrix-Vector Computation

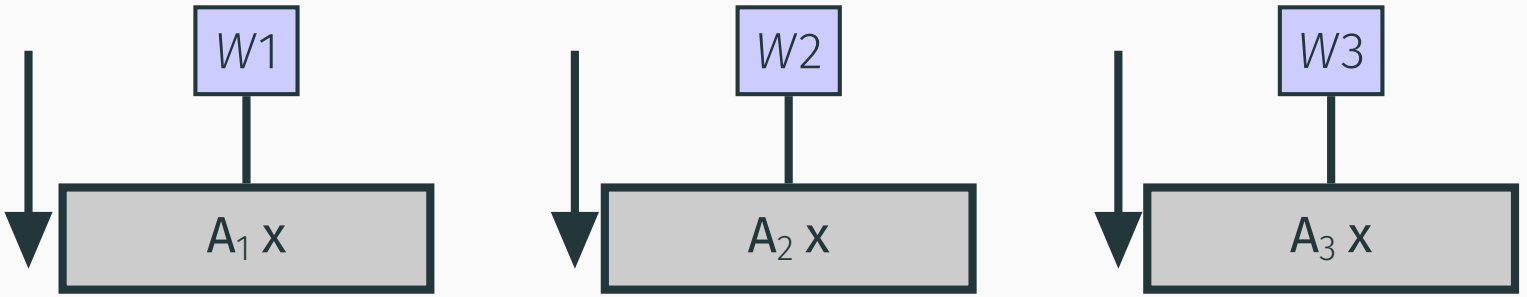


Block rows of A

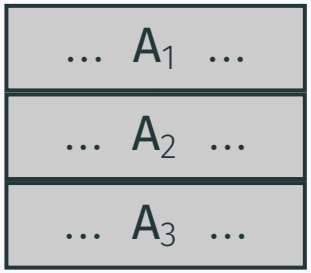
Distributed Matrix-Vector Computation



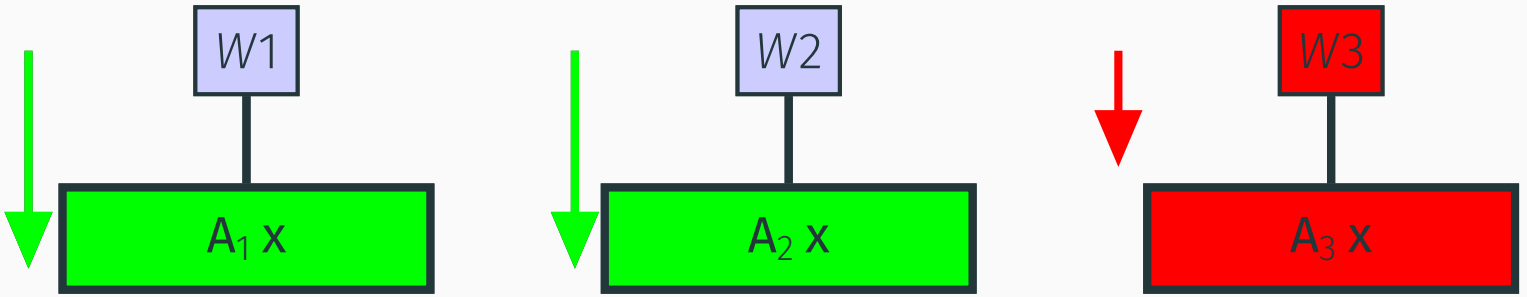
Block rows of A



Distributed Matrix-Vector Computation

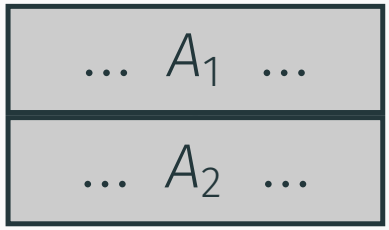


Block rows of A



Execution time dominated by the speed of the slowest worker.

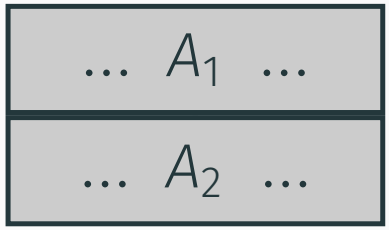
Coded Matrix-Vector Multiplication [Lee et al. ‘16]



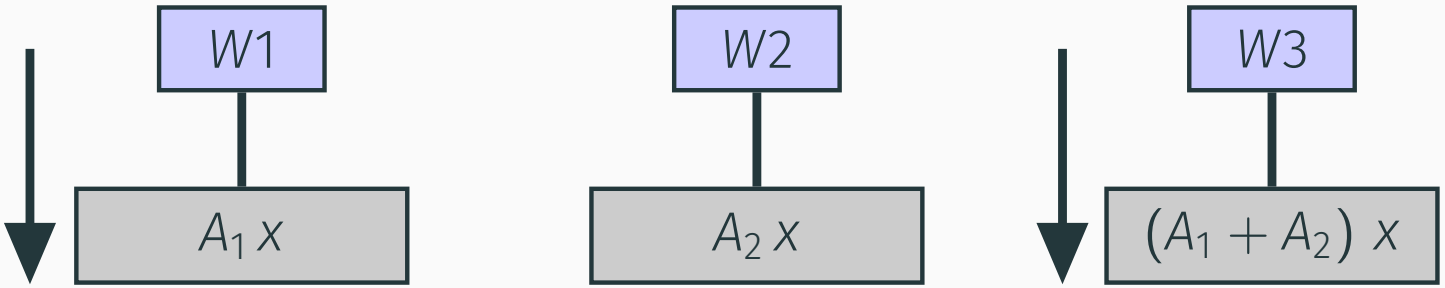
Matrix A

- Master node calculates $A_1 + A_2$ and sends A_1, A_2, and $A_1 + A_2$ and the vector x to the worker nodes.
- Master node can decode as long as any two worker nodes complete their tasks.

Coded Matrix-Vector Multiplication [Lee et al. '16]

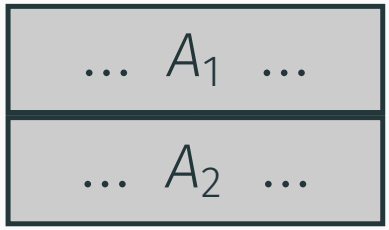


Matrix A

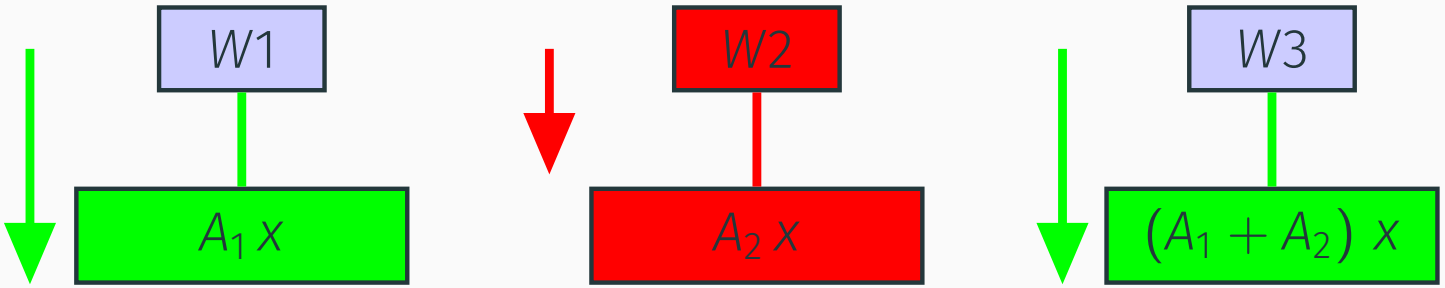


- Master node calculates $A_1 + A_2$ and sends A_1, A_2, and $A_1 + A_2$ and the vector x to the worker nodes.
- Master node can decode as long as any two worker nodes complete their tasks.

Coded Matrix-Vector Multiplication [Lee et al. '16]



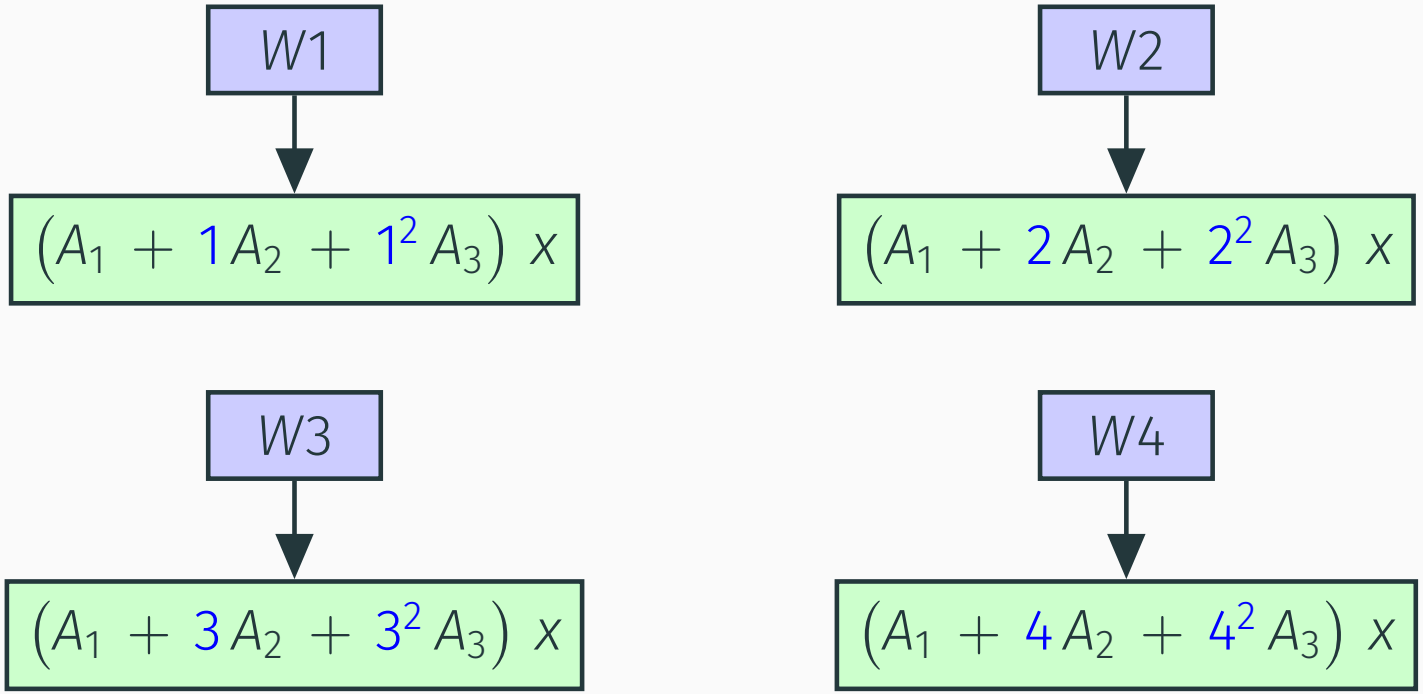
Matrix A



- Master node calculates $A_1 + A_2$ and sends A_1, A_2, and $A_1 + A_2$ and the vector x to the worker nodes.
- Master node can decode as long as any two worker nodes complete their tasks.

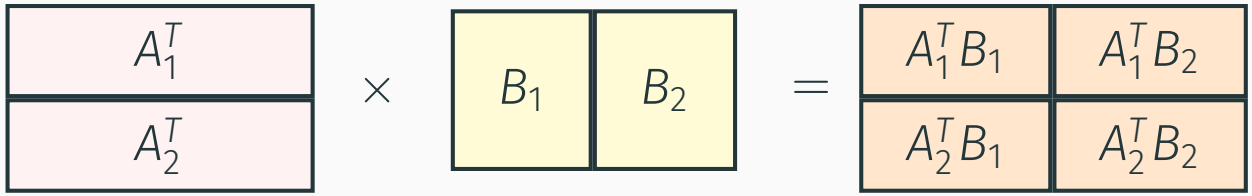
Coded Matrix-Vector Multiplication [Lee et al. '16]

- Natural generalization via Reed-Solomon-like approach.



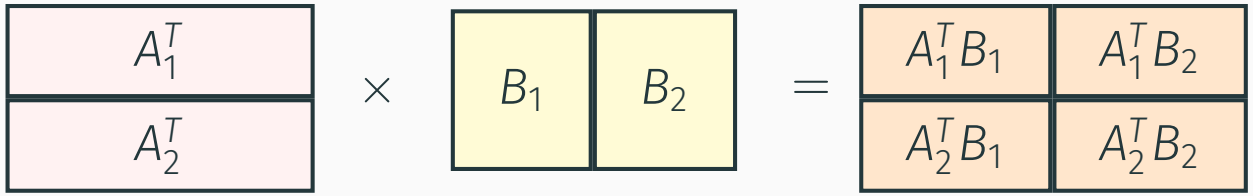
- Master node evaluates $A_1 + A_2 z + A_3 z^2$ at $z = 1, \ldots, 4$ and sends the evaluations and x to the workers.
- Result can be evaluated by polynomial interpolation at master node as long as at least *three* workers complete.

Coded Matrix-Matrix Multiplication [Yu et al. '17]



Matrix A^T Matrix B Product

Coded Matrix-Matrix Multiplication [Yu et al. '17]

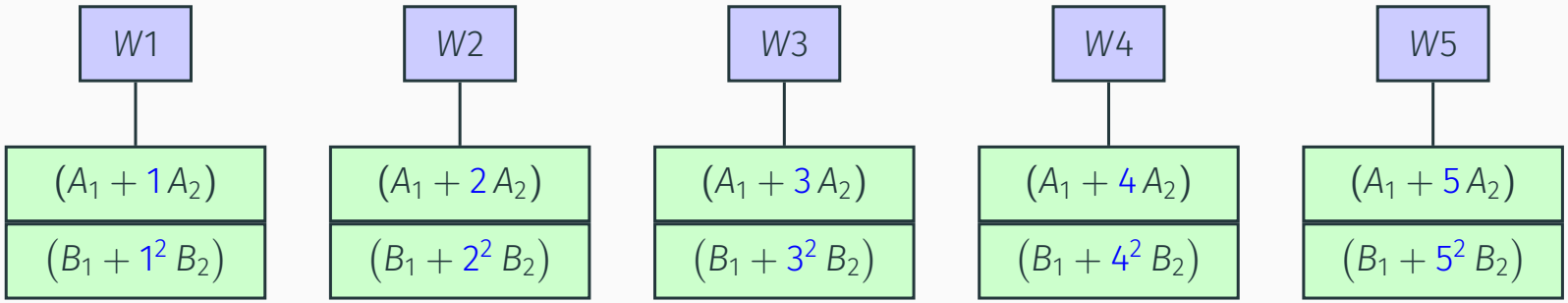


Matrix A^T Matrix B Product

Master node evaluates polynomials

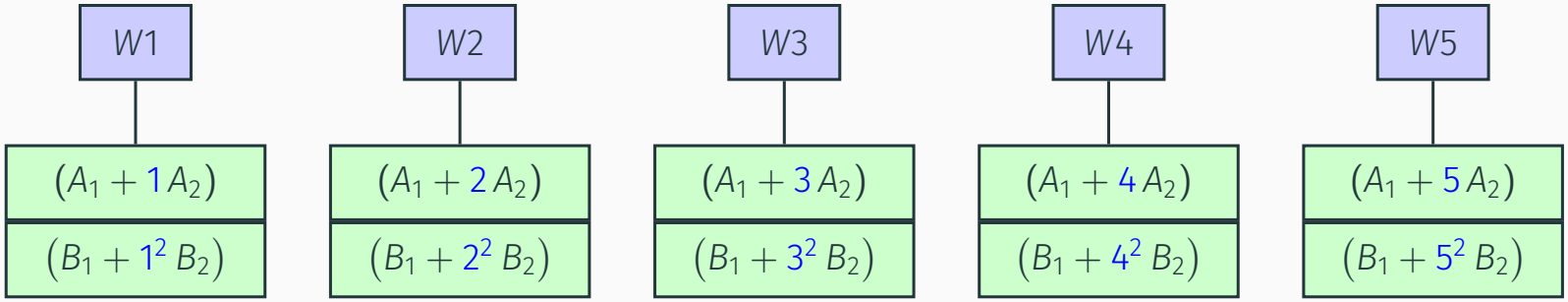
$A_1 + A_2 z$ and $B_1 + B_2 z^2$ at five different evaluation points.

Only requires scalar multiplication and addition.



Coded Matrix-Matrix Multiplication

Coded Matrix-Matrix Multiplication



Worker node i equivalently calculates

$$A_1^T B_1 + i\, A_2^T B_1 + i^2\, A_1^T B_2 + i^3\, A_2^T B_2$$

Degree sequence chosen carefully ...

Figures of merit for Coded Computation

- Coding for matrix computations essentially embeds the computation into a Reed-Solomon code.
- Schemes are clearly resilient to the maximum number of node failures.
  - Follows directly from RS-like structure.
- Recovery threshold τ is the minimum number of nodes that need to return their results to the master node for successful decoding.

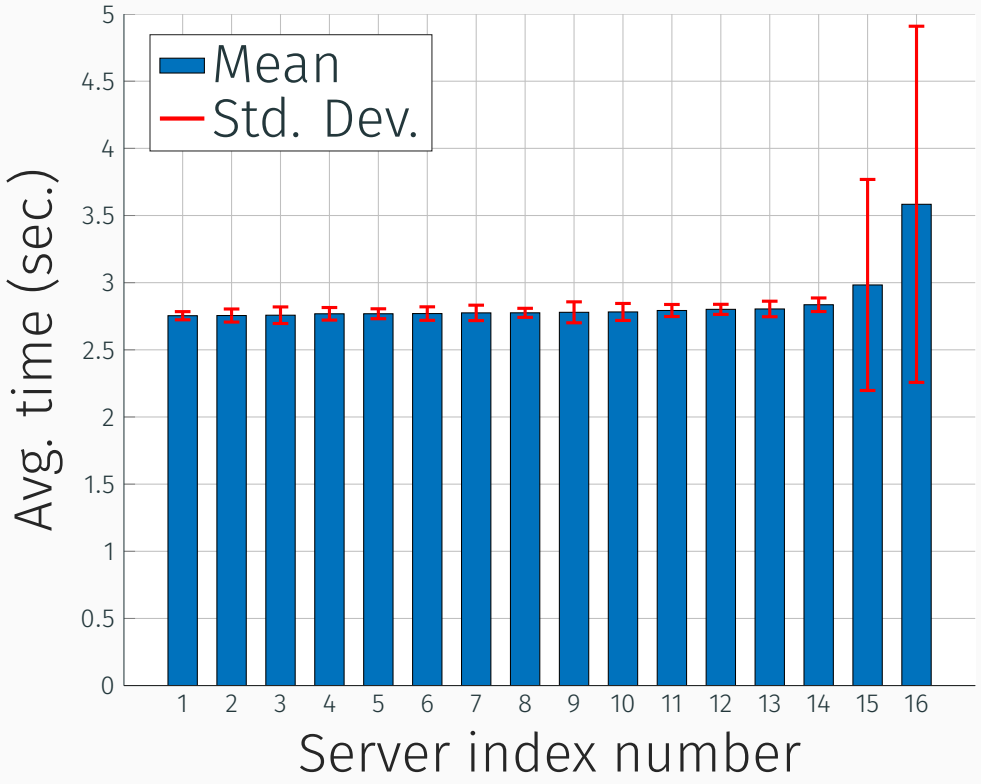
Issues with current approaches: Partial Stragglers

Stragglers are not the same as erasures ...

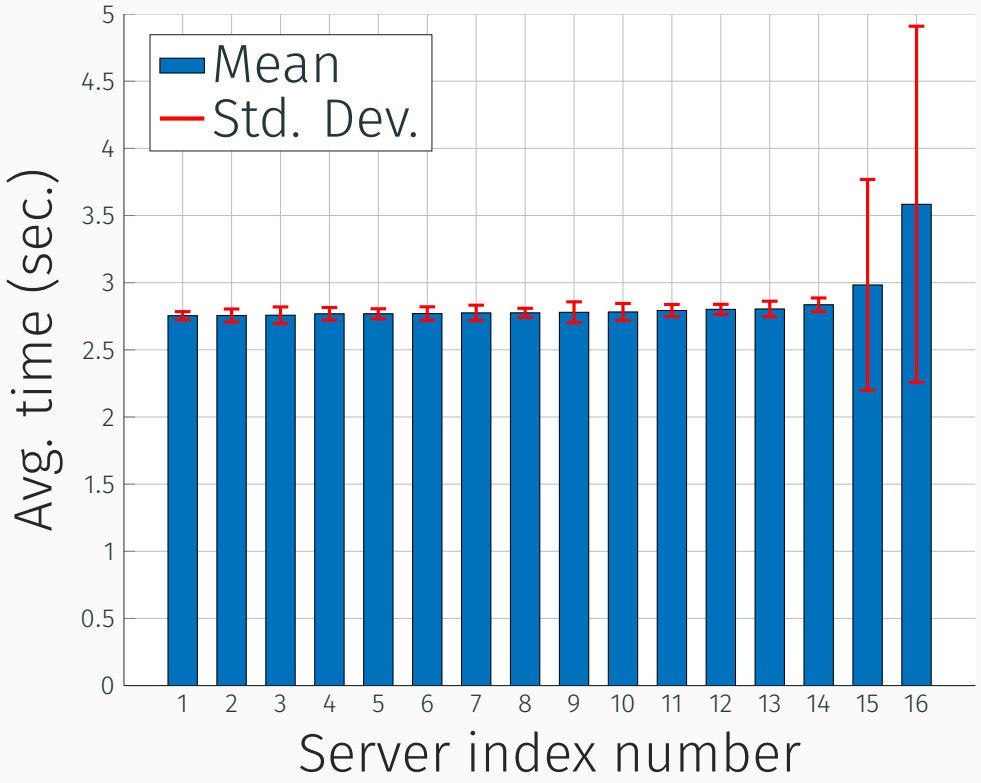
Unless they are complete node failures

Partial stragglers can be useful ...

Issues with current approaches: Partial Stragglers



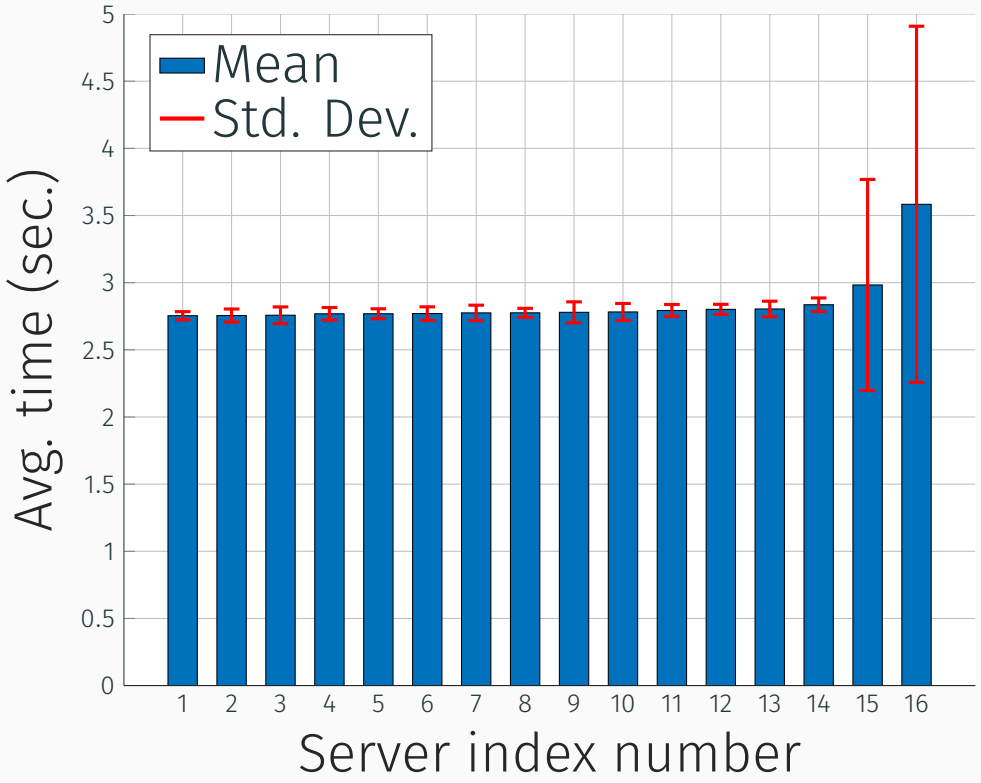
Issues with current approaches: Partial Stragglers



Approximately 10% of machines are slow stragglers, but not failures ...

On 16 AWS t2.micro machines

Issues with current approaches: Partial Stragglers



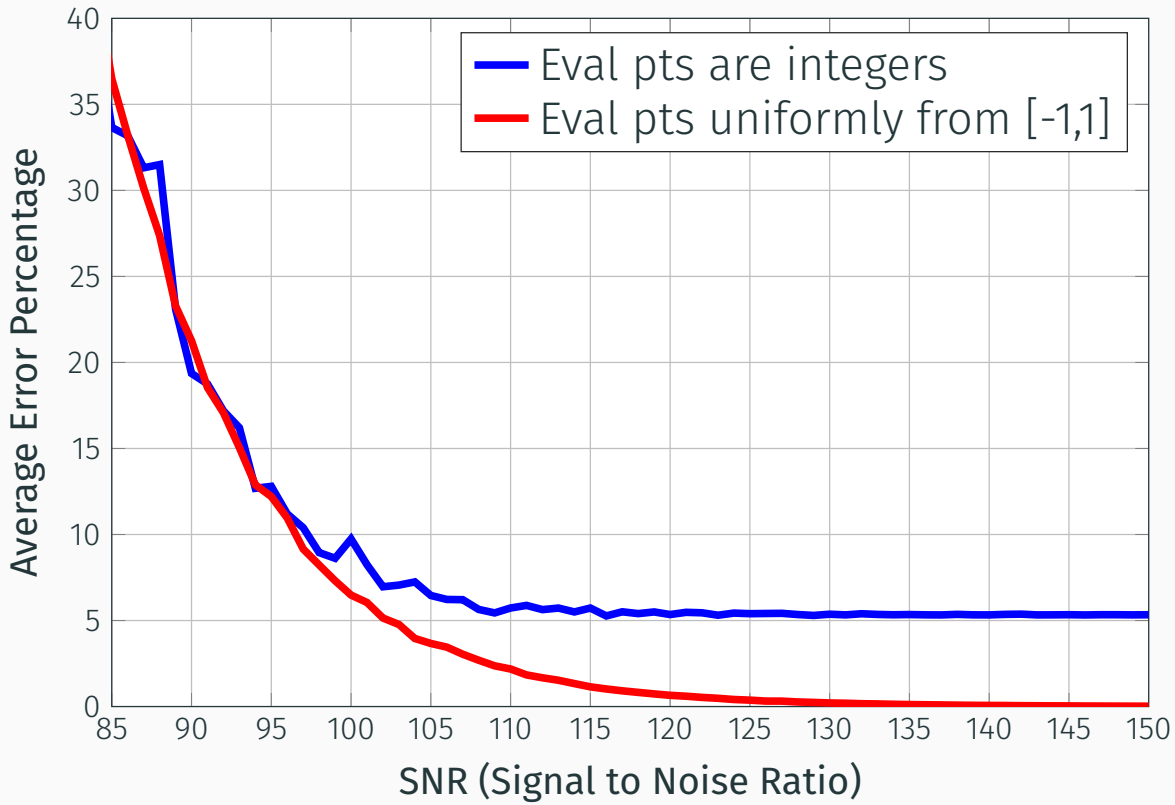
Modeling the speeds of different stragglers is not easy ...

Issues with current approaches: Numerical Stability

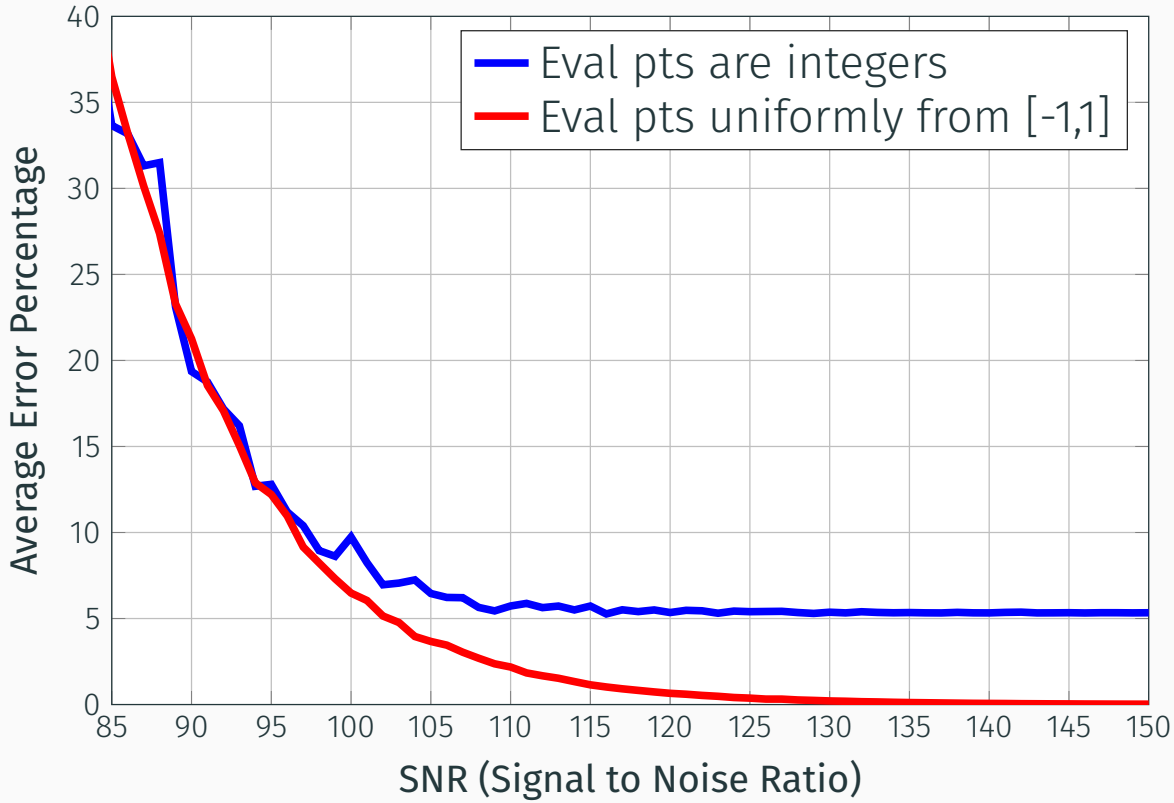
Vandermonde matrices have very bad condition numbers ...

Condition number of 10^{ℓ} $\approx$ loss of ℓ bits of numerical precision

Issues with current approaches: Numerical Stability



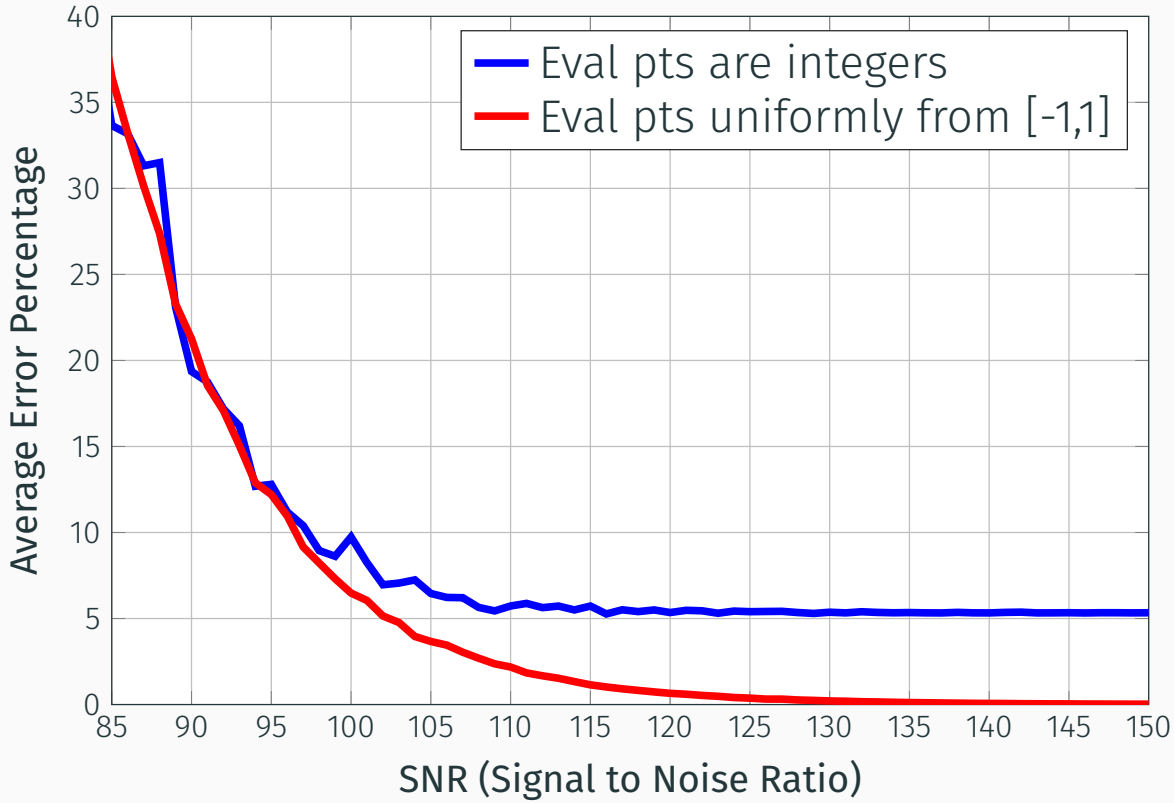
Issues with current approaches: Numerical Stability



Results of interpolating a noisy degree-9 polynomial.

Even at 100 dB, over 5% error …

Issues with current approaches: Numerical Stability



Problematic for machine learning applications ...

Gradient computations are often noisy.

Issues with curent approaches: Structured matrices

Many practical situations involve sparse matrices.

Issues with curent approaches: Structured matrices

Many practical situations involve sparse matrices.

Embedding into polynomial of deg-$(k-1)$ increases sparsity level k times.

Issues with curent approaches: Structured matrices

Many practical situations involve sparse matrices.

Embedding into polynomial of deg-$(k-1)$ increases sparsity level k times.

May even cause computation times to go up [Wang et al. '18] ...

A Fine Grained Model

We divide matrix A into Δ blocks, row-wise.

Each worker has a storage capacity of $\gamma = \frac{\ell}{\Delta}$.

Workers sequentially process blocks from top to bottom.

Computation is complete when *any* Q blocks are processed.

A Fine Grained Model

We divide matrix A into Δ blocks, row-wise.

Each worker has a storage capacity of $\gamma = \frac{\ell}{\Delta}$.

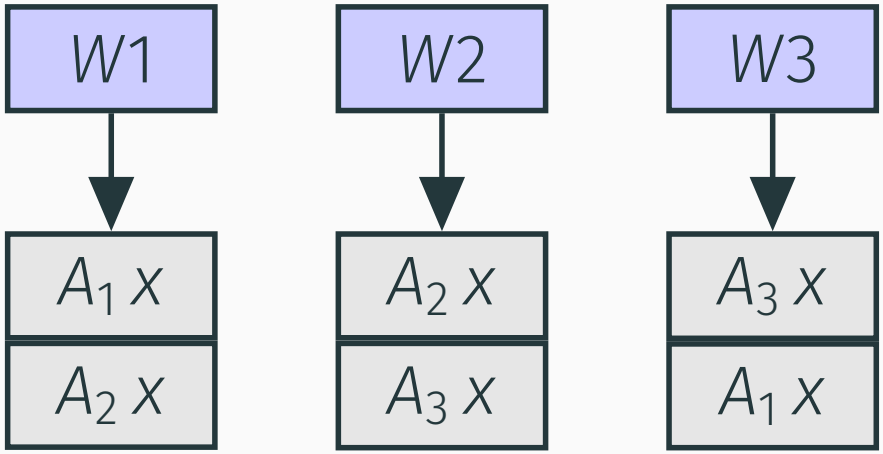
Workers sequentially process blocks from top to bottom.

Computation is complete when *any* Q blocks are processed.

Allows for simple way of capturing different worker speeds!

Ratio Q/Δ : worst-case computation that needs to take place.

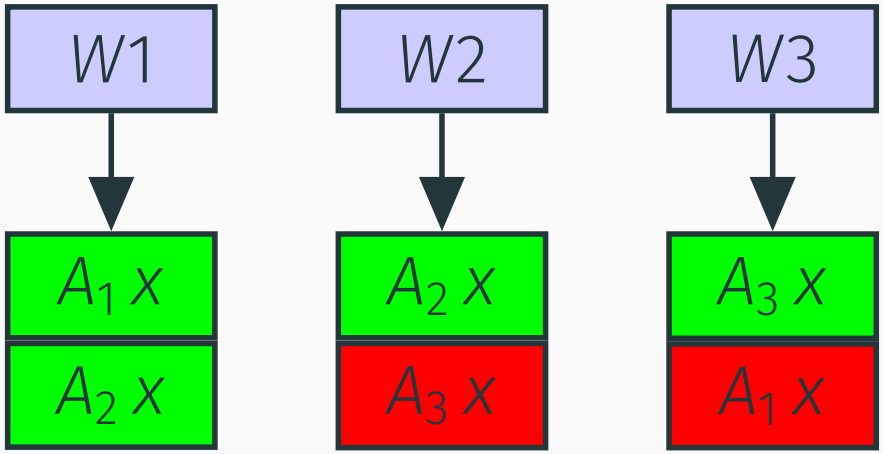
A Fine Grained Model



$\Delta = 3, \gamma = \frac{2}{3}, \mathbf{Q} = 4$

Uncoded Solutions

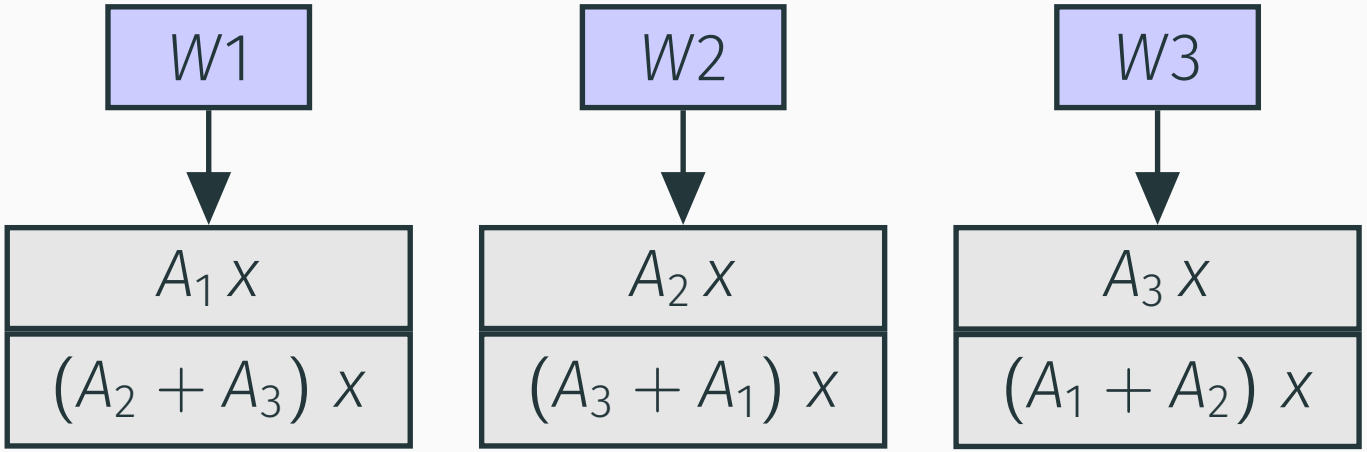
A Fine Grained Model



$\Delta = 3, \gamma = \frac{2}{3}, \mathsf{Q} = 4$

Uncoded Solutions

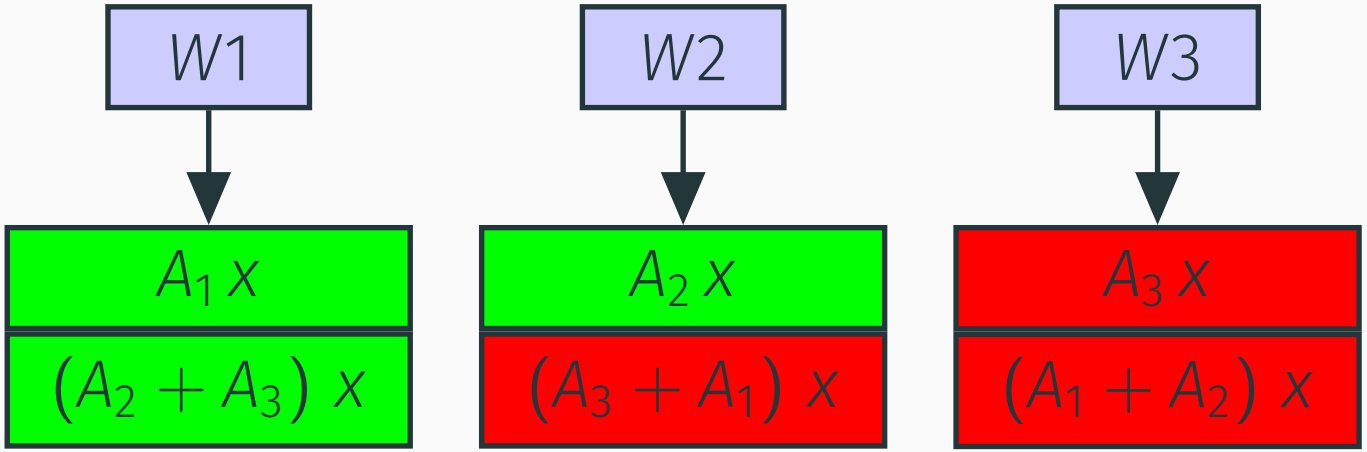
A Fine Grained Model



$$\Delta = 3, \gamma = \tfrac{2}{3}, \mathbf{Q} = 3$$

Partially Coded Solution

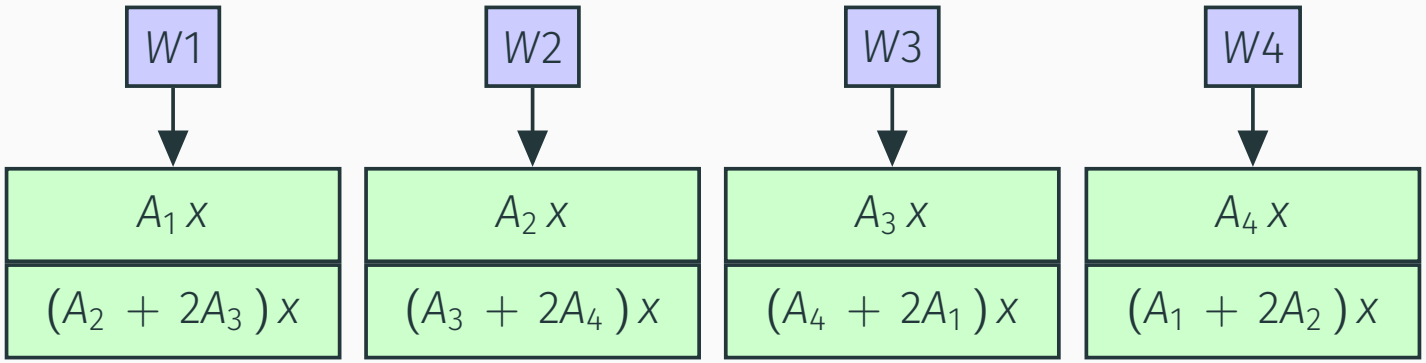
A Fine Grained Model



$$\Delta = 3, \gamma = \tfrac{2}{3}, \mathbf{Q} = 3$$

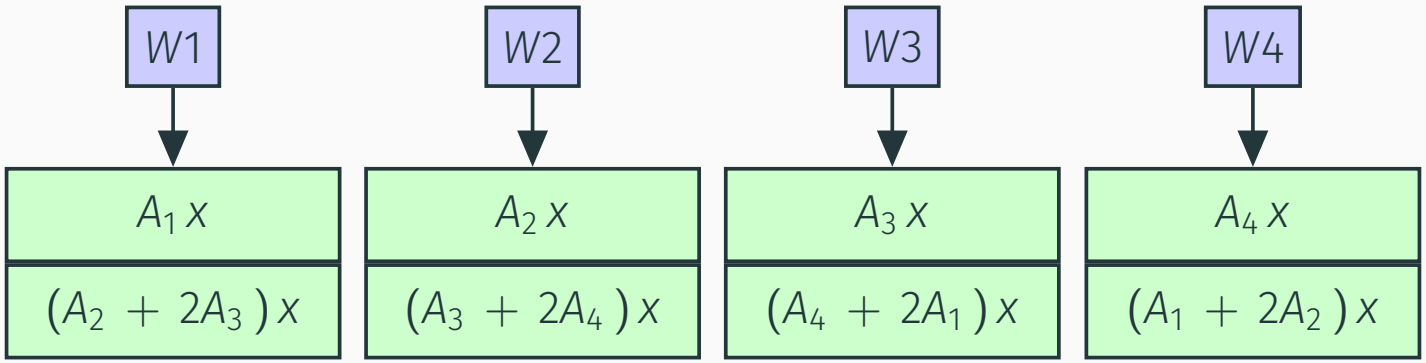
Partially Coded Solution

Ordering of blocks matters!



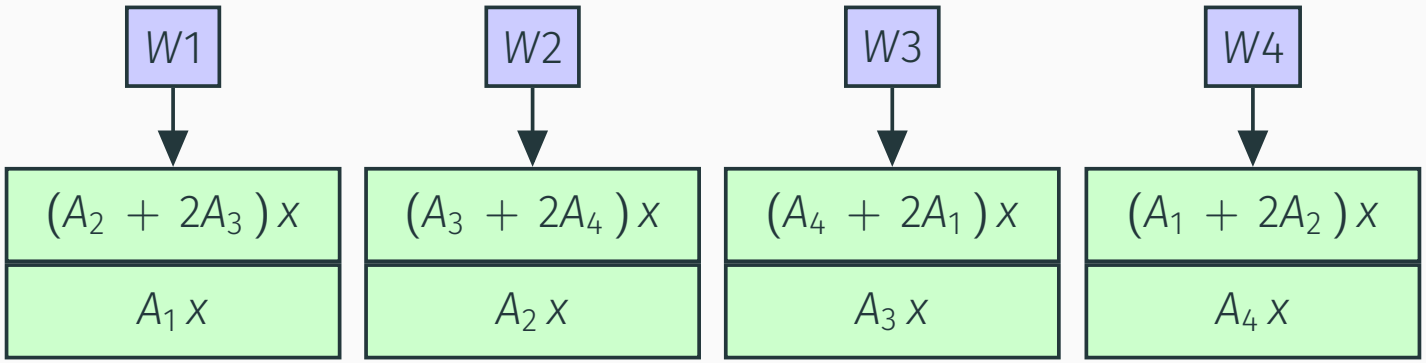
A scheme with Q $= 5$

Ordering of blocks matters!



A scheme with Q $= 5$

Placing the coded blocks first, reduces **Q** ...



A scheme with Q $= 4$

Dealing with sparsity & numerical stability issues

Constrain the fraction of coded blocks in each worker

Uncoded fraction γ_u

Coded fraction γ_c

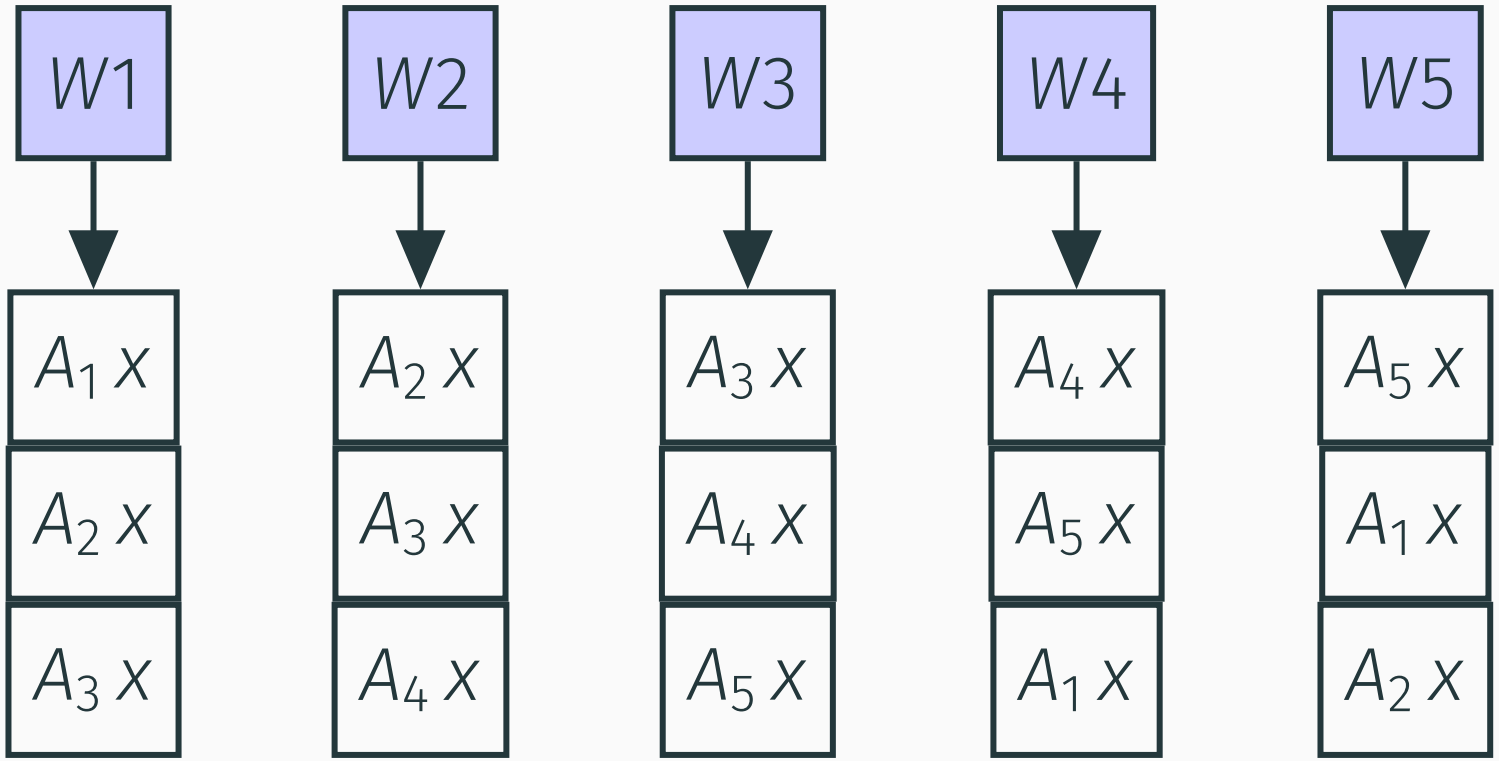
Key Questions under consideration

For the fine grained model

Bounds on Q/Δ?

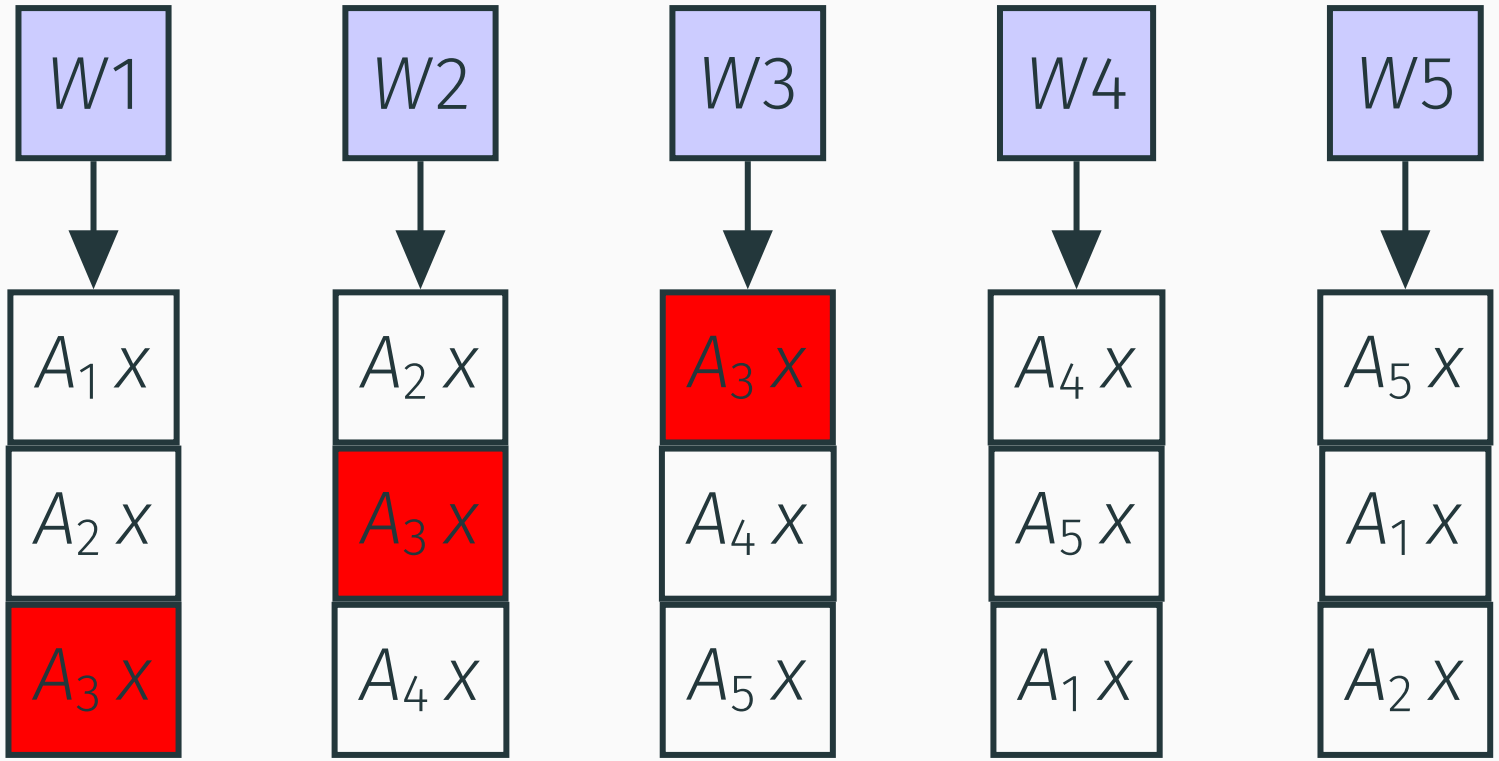
Achievability schemes for Q/Δ?

Uncoded Scheme



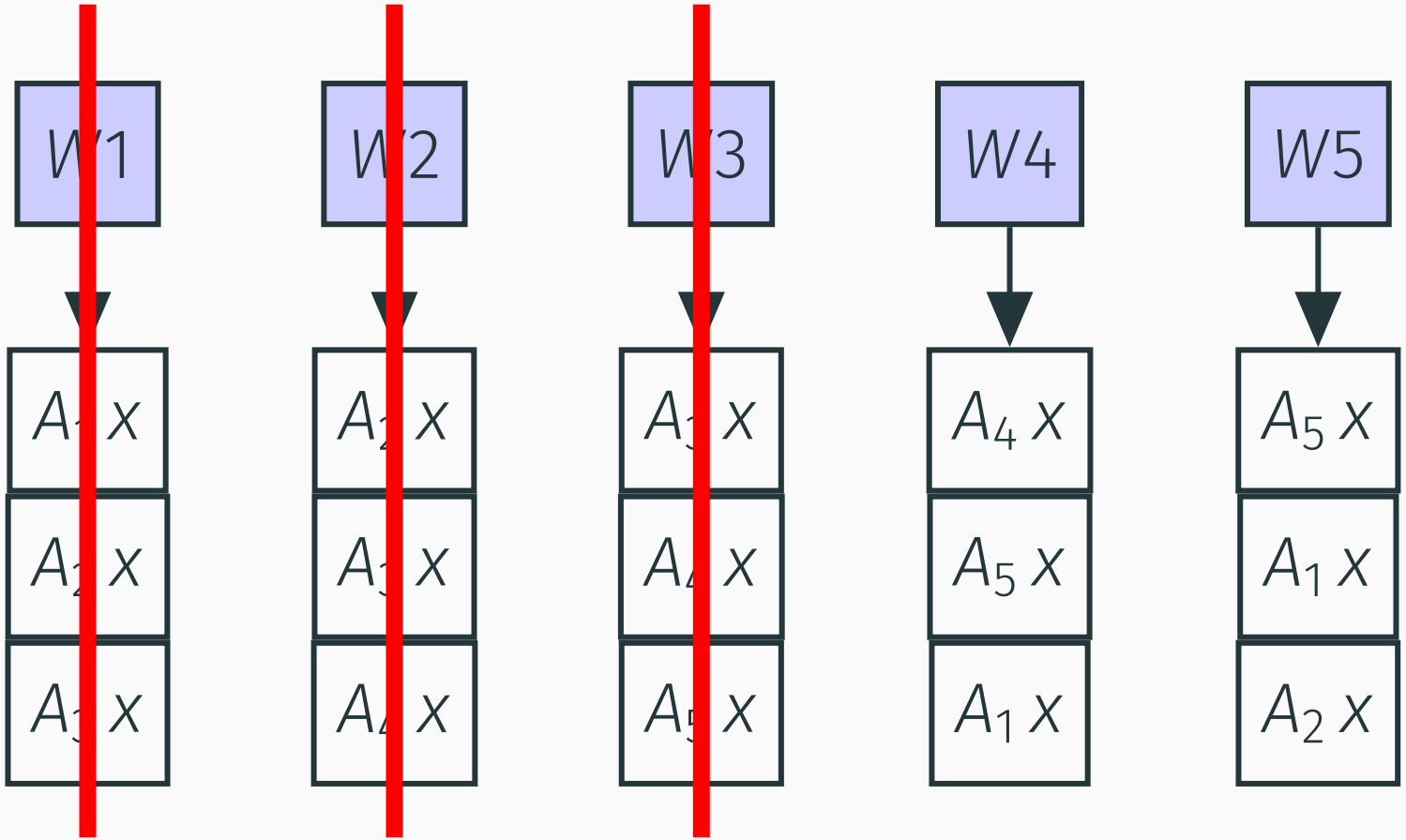
All blocks are uncoded ...

Uncoded Scheme



All blocks are uncoded ...

Uncoded Scheme



Definitely need one copy of each uncoded block ...

Uncoded Scheme: Failure resilience

- Let r be the number of occurrences of each block. Then

$$n\ell = r\Delta$$

Uncoded Scheme: Failure resilience

- Let r be the number of occurrences of each block. Then

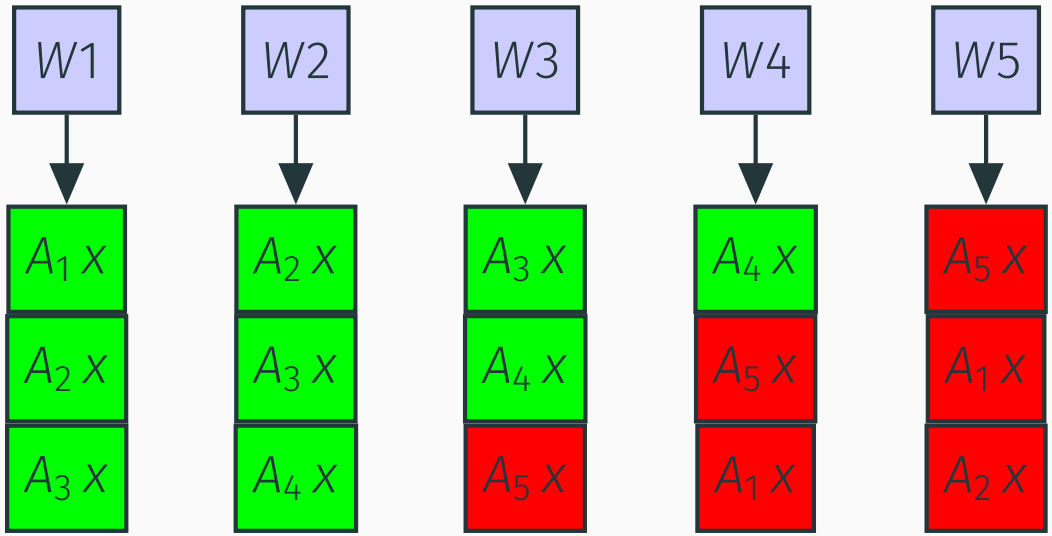
$$n\ell = r\Delta$$

Failure resilience

Consider an $\langle n, \ell, \Delta, r \rangle$-uncoded system. If the system needs to be resilient to s stragglers, then $r \geq s + 1$ and $n\gamma = r$.

Uncoded Scheme: Lower bound on Q/Δ

Example: $Q_5 = 9$



- At least one copy of each $A_j x$ needs to be obtained by the master node.
- Q_j : number of blocks processed in the worst case without processing $A_j x$.

Uncoded Scheme: Lower bound on Q/Δ

$$Q = 1 + \max_{j=1,\ldots,\Delta} Q_j$$

Basic Averaging argument yields

$$Q \geq 1 + \frac{\sum_{j=1}^{\Delta} Q_j}{\Delta}$$

Uncoded Scheme: Lower bound on Q/Δ

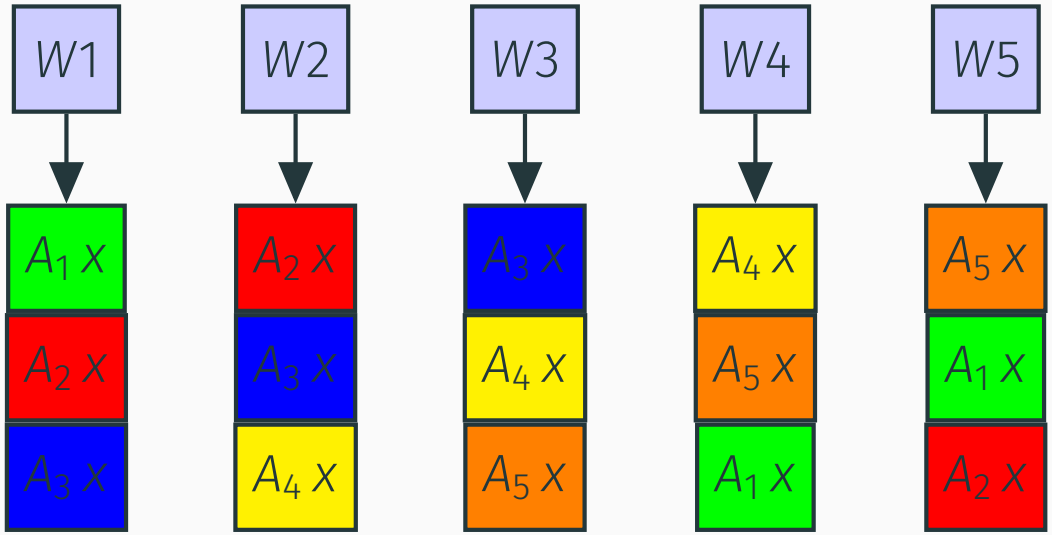
Combinatorial argument: Counting $\bar{Q}$ two ways ...

Uncoded Q/Δ bound

In an $\langle n, \ell, \Delta, r \rangle$-uncoded system, $Q \geq \max(\Delta, \Delta r - \frac{r}{2}(\ell + 1) + 1)$.

Uncoded Scheme: Matching Constructions Exist

Choosing $\Delta = n$ and placing blocks in a cyclic shift manner ..



$\Delta = n = 5, \ell = 3$

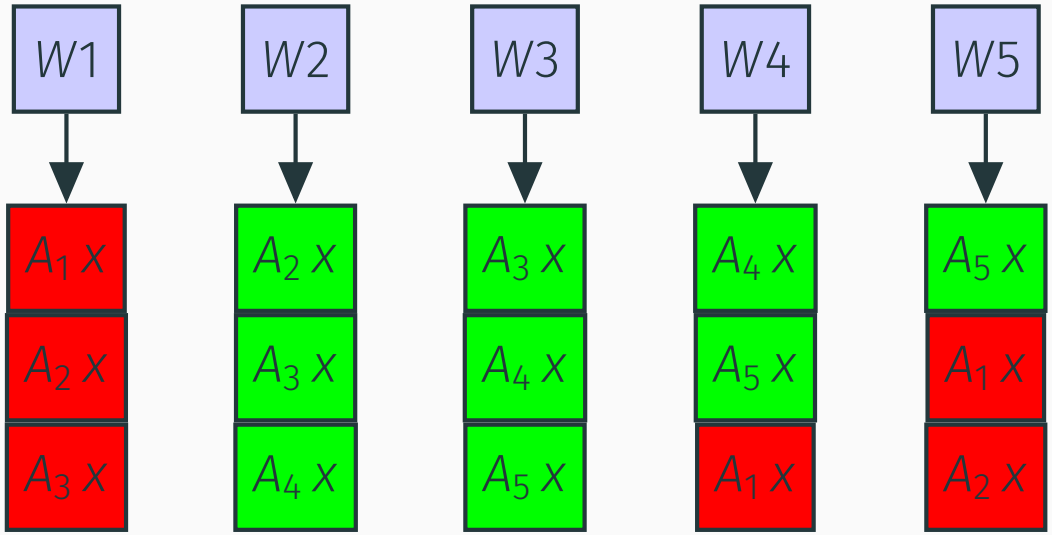
Resilient to two failures and meets the $Q/\Delta = 2$ bound.

Uncoded Scheme: Matching Constructions Exist

Choosing $\Delta = n$ and placing blocks in a cyclic shift manner ..

Uncoded Scheme: Matching Constructions Exist

Choosing $\Delta = n$ and placing blocks in a cyclic shift manner ..

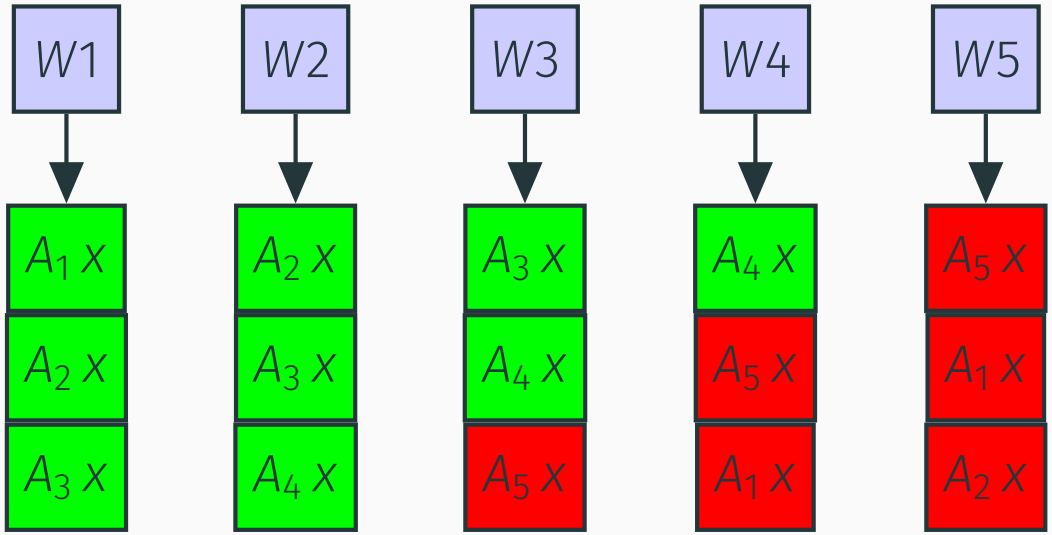


$\Delta = n = 5, \ell = 3$

$Q_1 = 9$

Uncoded Scheme: Matching Constructions Exist

Choosing $\Delta = n$ and placing blocks in a cyclic shift manner ..



$\Delta = n = 5, \ell = 3$

$Q_5 = 9$

Uncoded Scheme: Matching Constructions Exist

Choosing $\Delta = n$ and placing blocks in a cyclic shift manner ..

With cyclic construction Q_j is the same for each j!

Mix of uncoded and coded blocks

- Depends on where the coded and uncoded symbols appear in the worker nodes.
- If we expect stragglers to be somewhat infrequent, then it makes sense to put uncoded on top and coded blocks at the bottom. Most of the time decoding complexity will be low.
- Placing coded blocks at the top may reduce the worst-case computation at the expense of decoding complexity.

Mix of coded and uncoded blocks

- We now assume that each node receives a $\gamma = \gamma_u + \gamma_c$ fraction of the rows of A, where γ_u and γ_c correspond to the uncoded and coded parts respectively.

- A $\langle n, \ell_u, \ell_c, \Delta, r_u \rangle$ system means
  - n workers, ℓ total blocks in a worker: ℓ_u-uncoded and ℓ_c-coded.
  - Each block appears r_u times in the uncoded part.

Coded Blocks at the Bottom: Lower Bound on $\frac{Q}{\Delta}$

In the best case each coded block is "useful" to the master node

Coded blocks at bottom Q/Δ bound

Consider a $\langle n, \ell_u, \ell_c, \Delta, r_u \rangle$ system with coded blocks at the bottom. Then,

$$Q_{cb} \geq \max(\Delta, \Delta r_u - \frac{r_u}{2}(\ell_u + 1) + 1)$$

. Furthermore, it is resilient to $\left\lfloor \frac{n^2\gamma_c + n\gamma_u - 1}{n\gamma_c + 1} \right\rfloor$ failures.

Lower bound follows from uncoded bound discussed earlier ...

Coded Blocks at the Bottom: Lower Bound on $\frac{Q}{\Delta}$

In the best case each coded block is "useful" to the master node

Coded blocks at bottom Q/Δ bound

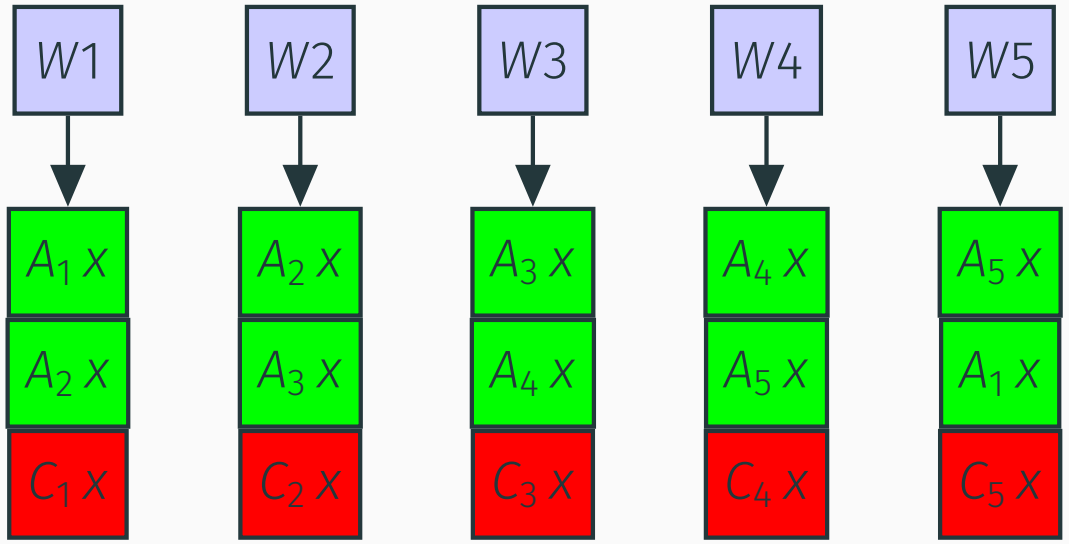
Consider a $\langle n, \ell_u, \ell_c, \Delta, r_u \rangle$ system with coded blocks at the bottom. Then,

$$Q_{cb} \geq \max(\Delta, \Delta r_u - \frac{r_u}{2}(\ell_u + 1) + 1)$$

. Furthermore, it is resilient to $\left\lfloor \frac{n^2\gamma_c + n\gamma_u - 1}{n\gamma_c + 1} \right\rfloor$ failures.

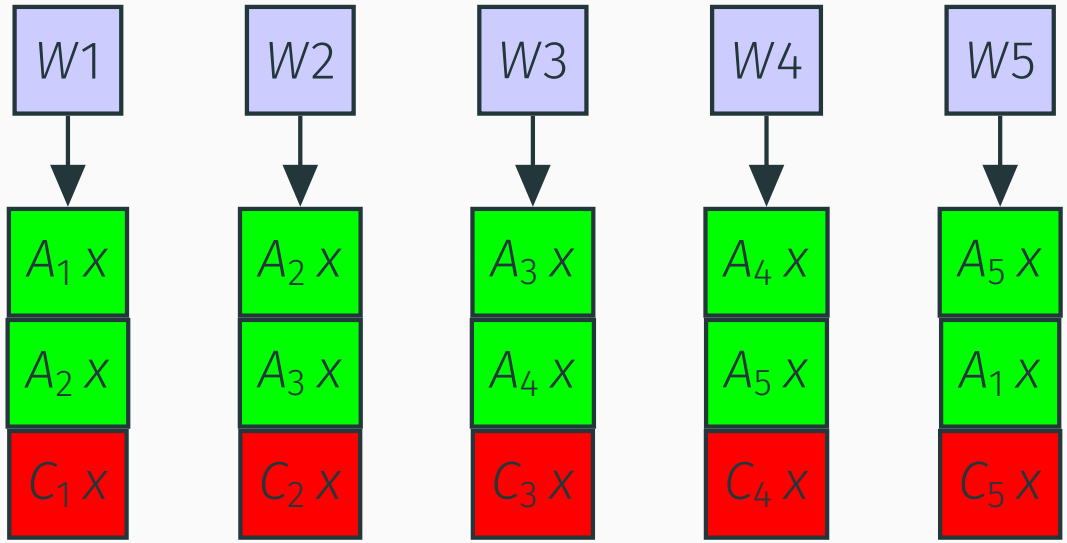
Lower bound follows from uncoded bound discussed earlier ...

Coded Blocks at the Bottom: Matching Construction

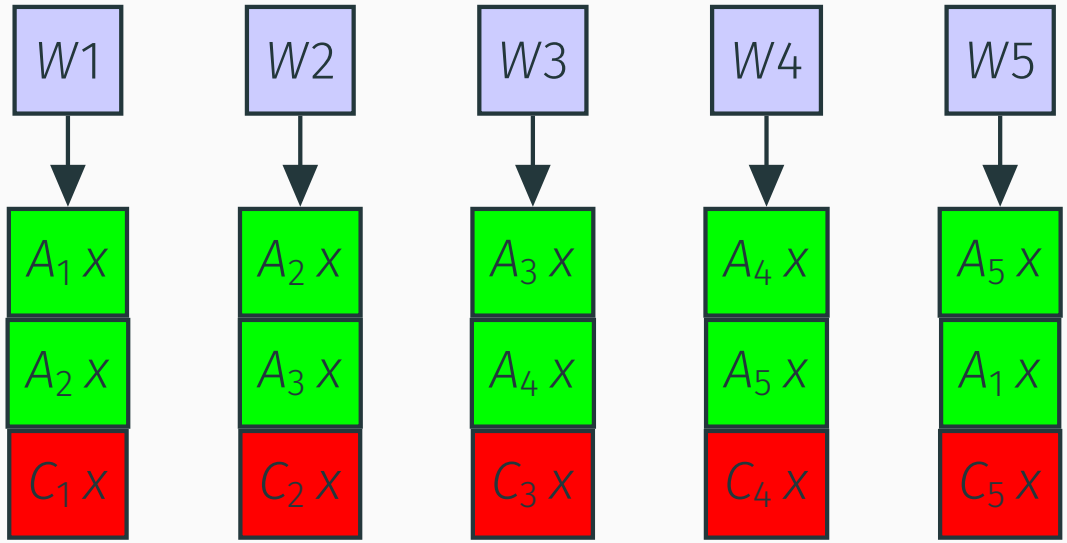


- We consider the same scenario with $n = 5$ and $\gamma = \frac{3}{5}$
- The scheme is resilient to 3 stragglers and $Q_{cb} = 8$ if $\ell_u = 2$.

Coded Blocks at the Bottom: Matching Construction



Coded Blocks at the Bottom: Matching Construction



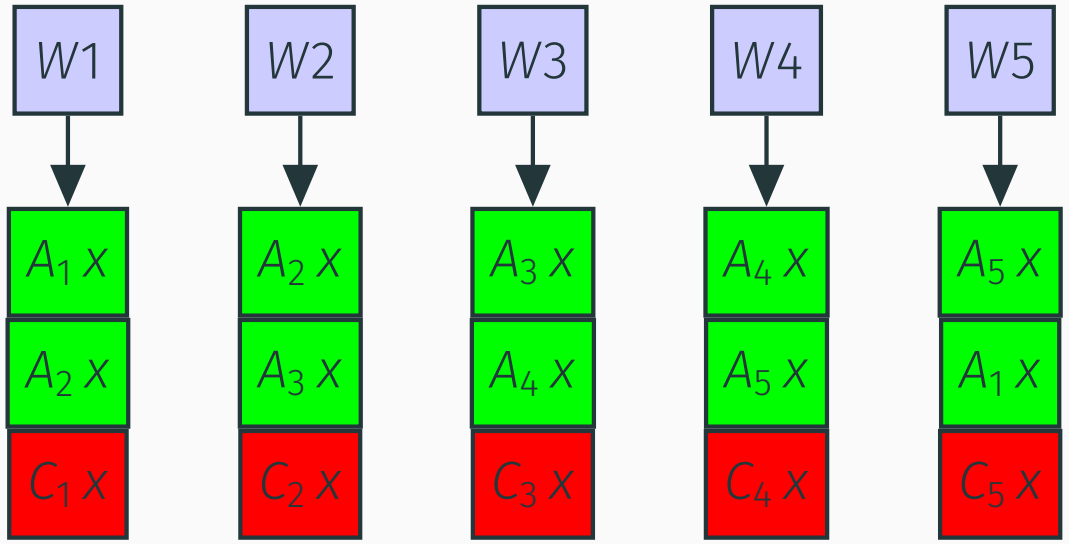
Requires the usage of Cauchy matrices for the coded blocks.

Cauchy

An $m \times n$ matrix with elements $a_{ij} = \frac{1}{x_i - y_j}$ where x_i and y_j are sequences of distinct elements, where $x_i \neq y_j$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

Any square submatrix is full-rank.

Coded Blocks at the Bottom: Matching Construction



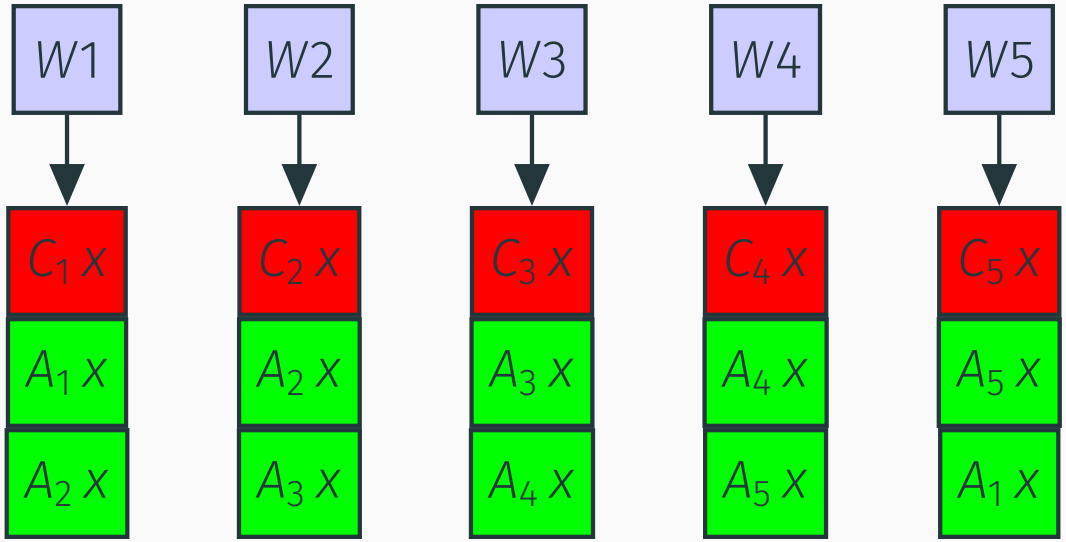
Basic Idea: Place uncoded blocks in a cyclic manner at the top

Cauchy matrices of appropriate dimension at the bottom.

Coded Blocks at the Top

- A given worker node only processes uncoded blocks after having processed ℓ_c coded blocks.
- If x coded blocks are processed by workers, then it suffices if *any* $\Delta - x$ blocks are processed in the uncoded part
  - The *any* $\Delta - x$ fact makes things a lot harder.
- Have lower bounds, but not matching constructions in general ...

Coded Blocks at the Top



- Consider scenario where $\Delta = n = 5$ and $\ell = 3$. Here, $\ell_u = r_u = 2$ and $\ell_c = 1$.
- Once again, lower bound corresponds to case where each coded symbol is useful.

Coded Blocks at the Top: Lower bound on $\frac{Q}{\Delta}$

- Consider an arbitrary set of β worker nodes that process all their blocks and another set of worker nodes that only contribute x coded blocks.
- The total number of coded blocks is $x + \ell_c\beta$. Let $\mathcal{A}$ denote the set of distinct uncoded blocks from those β workers.

$$Q_{ct} \geq x + \ell\beta + 1, \text{ when}$$
$$x + \ell_c\beta + |\mathcal{A}| < \Delta.$$

  This is because, we do not have enough equations to decode the $\Delta - |\mathcal{A}|$ unknowns.

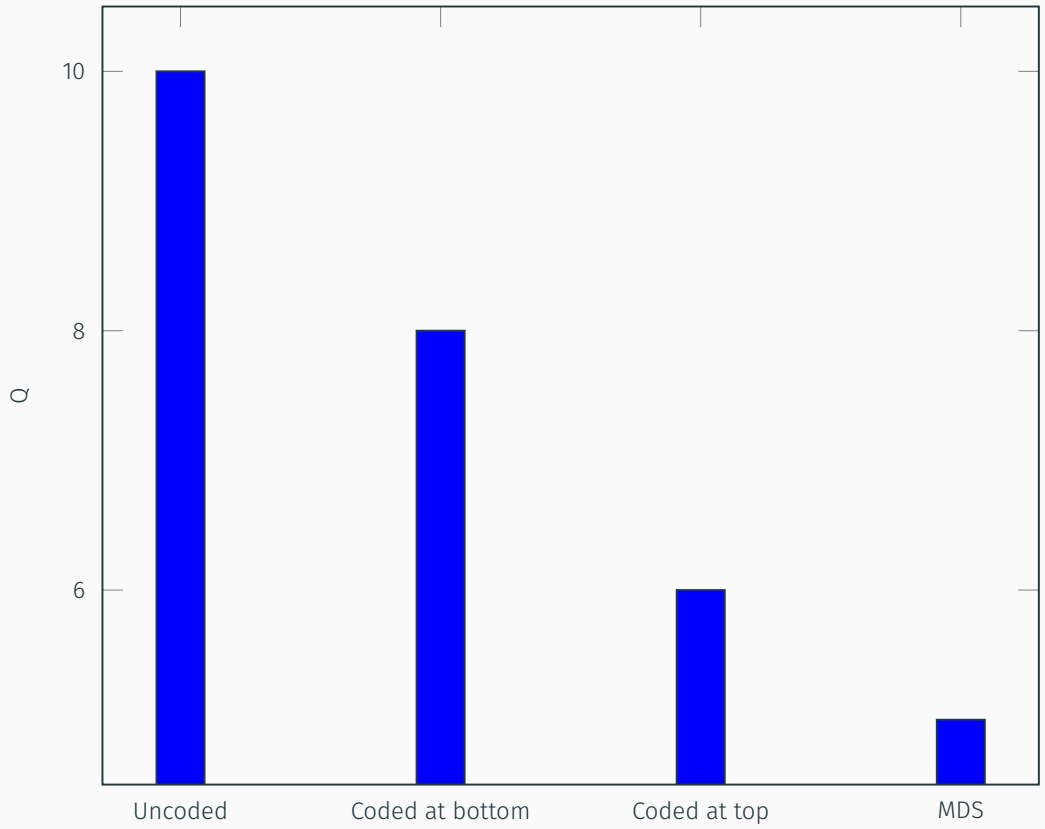
- Next, we use another averaging argument. We calculate the average size of $\mathcal{A}$ considering all possible $\binom{n}{\beta}$ worker nodes.

Coded Blocks at the Top

- A lower bound on Q_{ct} can be derived by solving the following optimization problem.

$$\begin{aligned} &\text{maximize} && x + \ell\beta + 1 \\ &\text{subject to} && (x + \ell_c\beta) < \Delta \left[\frac{\binom{n-r_u}{\beta}}{\binom{n}{\beta}} \right] \end{aligned}$$

Summary of Bound Examples



Q decreases as more coding is introduced.

Matching constructions in two of the cases.

Questions?