STRADS: A Distributed Dynamic Scheduler for Parallel Machine Learning

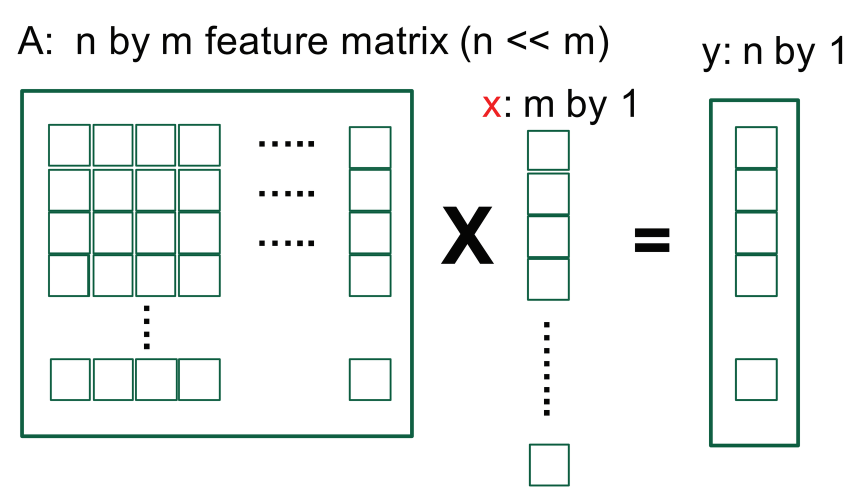
Jin Kyu Kim, Seunghak Lee, Eric Xing, Garth Gibson (CMU)

MOTIVATION

- Machine learning techniques are mostly defined serially
- Big Data drives ML to seek out parallel algorithms
  - Property: large # of features (dimensions)
  - Examples: Sensor array, consumer preference, Netflix

WHAT'S REGRESSION?

- Regression problem: For given input A, and observation Y, find unknown x parameter



Ex.: Alzheimer Disease data 463 sample by 509K features. In case of pair-wise study, # of features would be inflated to 10^{11}.

- Sparse regression is one variation of regression that favors a small number of non-zero parameters corresponding to the most relevant features

EVALUATION OF PREVIOUS WORKS

Sequential Lasso

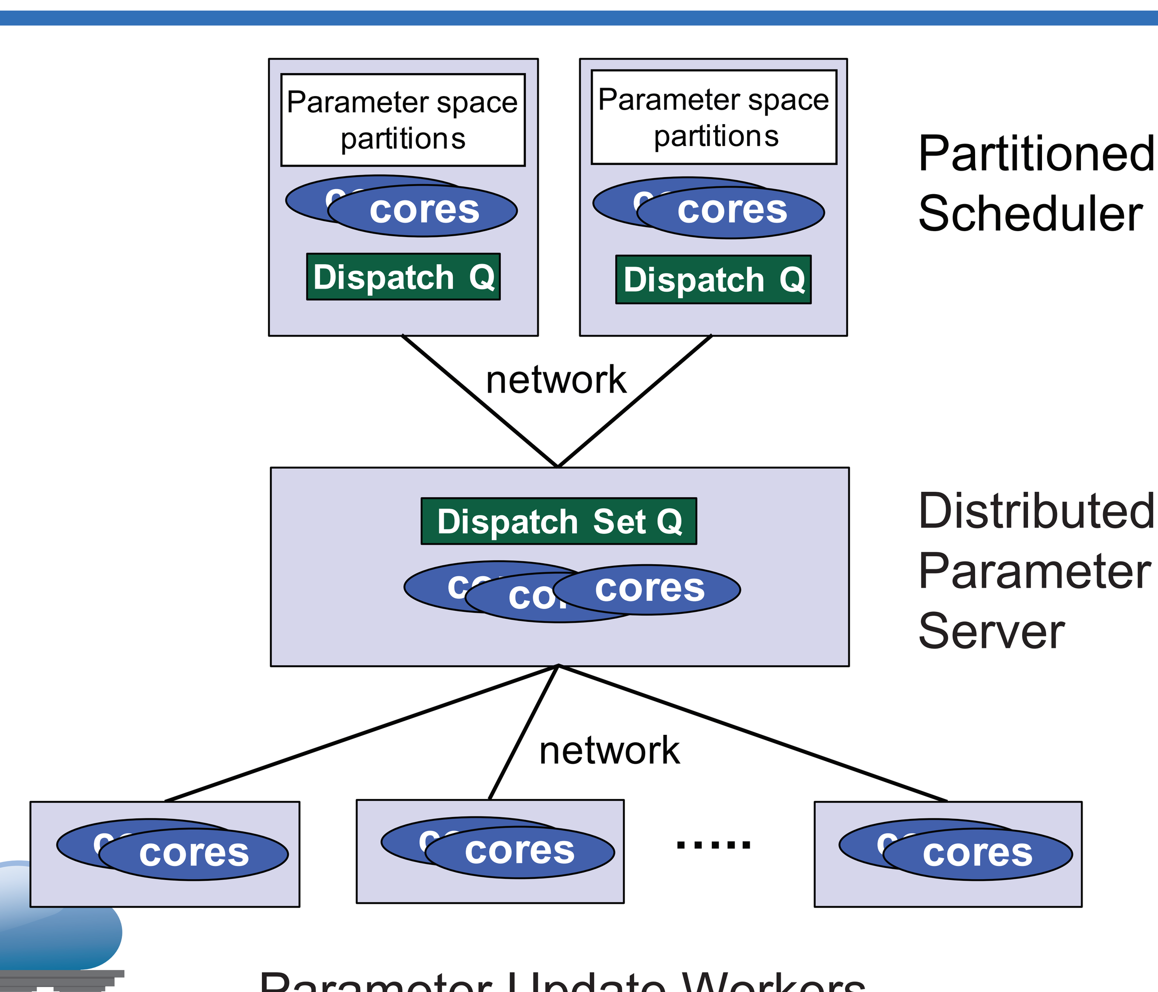
```
While(converge?){
 for(i=0; i<N; i++){
   X[i] = Update(i, X);
 }
calculate_objective_value(X);
}
```

Parallel Lasso (Shotgun)

```
While(converge?){
 for(i=0; i<N/P; i++){
   Choose P parameters in random
   Update P parameters in parallel.
   barrier()
   Update P entries in X
 }
 calculate_objective_value(X)
}
```

- Problems with previous work
  - Uniform scheduling: wastes most of cycles updating already-converged parameters
  - Random selection: limits the parallelism because of the risk of divergence or slow down

STRADS: IMPLEMENTATION STRATEGY

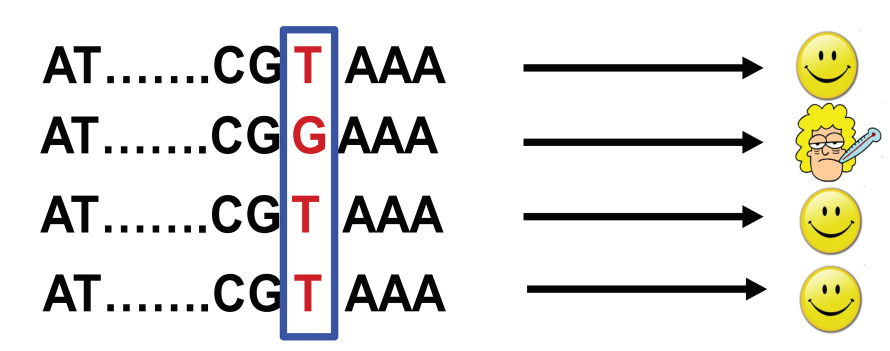


GOAL AND IDEAS

- Our goal: provide a faster way of solving high-dimension problems in parallel
- Approach: Application-aware task scheduler specific to convex optimization solving.
- We apply our two application-aware scheduling policies to "Sparse Regression Problem" as an example

"LASSO" FOR SPARSE REGRESSION

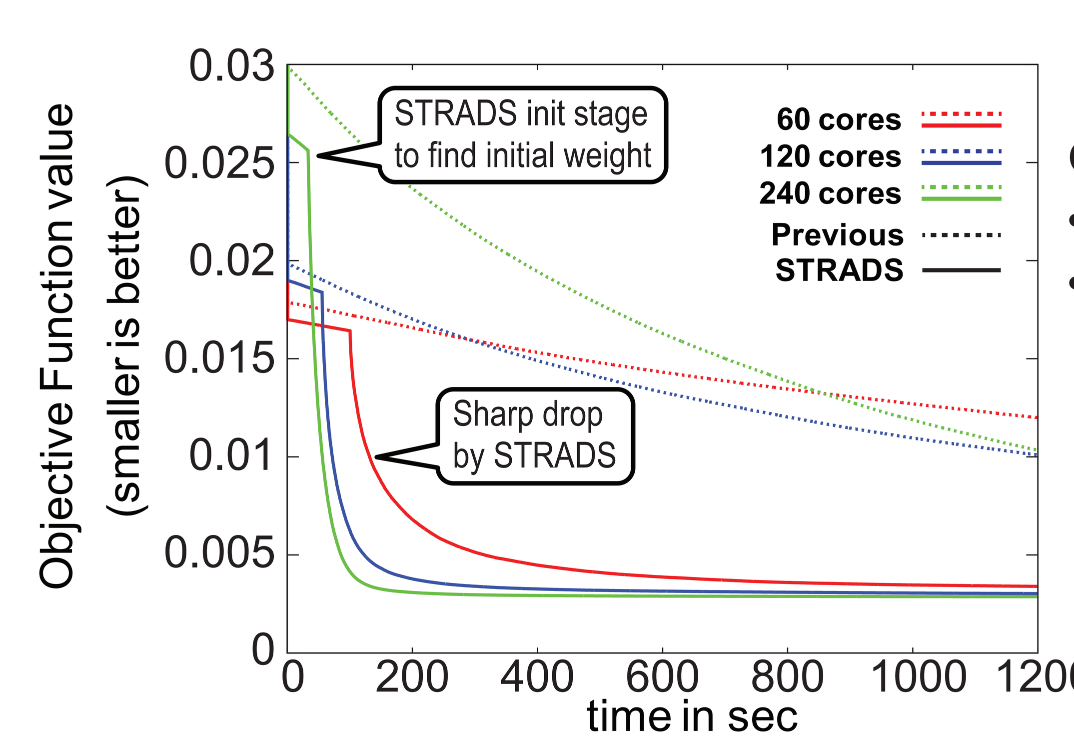
- Very famous algorithm for feature selection
  - Tibshirani '96 (8744 reference counts in Google scholar)
- Select features relevant to output (Y)
- Example: What gene affects cancer susceptibility?



STRADS: TWO SCHEDULING POLICIES

- First: Weight Distribution Based Sampling
  - Assign higher weight to more promising parameter update
  - Select parameters to update in parallel based on weighted coin (Δx from last update)
  - Result: important parameters updated more frequently
  - Improves convergence speed substantially
- Second: Run time error control
  - Estimate potential interference for pairs of parameters selected during sampling
  - If interference too strong, drop one parameter from parallel update set
  - Result: potential error is under control

EXPERIMENT AND CONCLUSIONS



Genetic Simulation Data
- 450 patients
- 1M genetic features

- Combination of two scheduling policies improves convergence rate substantially for sparse regression
- We believe that scheduling idea can be applied to other optimization solvers

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley · University of Washington