A Reliable Effective Terascale Linear Learning System

Alekh Agarwal Olivier Chappelle Miroslav Dudík John Langford

U C Berkeley

Yahoo! Research

What we do

- Billions of examples, millions of parameters, 1K nodes, precious little time
- Want to learn a good predictor of the form: $f_w = \sum_{i=1}^{p} w_i x_i$
- Subsampling smaller dataset hurts learning performance
- Example: Ad-click prediction problem

  2.1T non-zero features
  17B samples
  16M parameters
  1K nodes

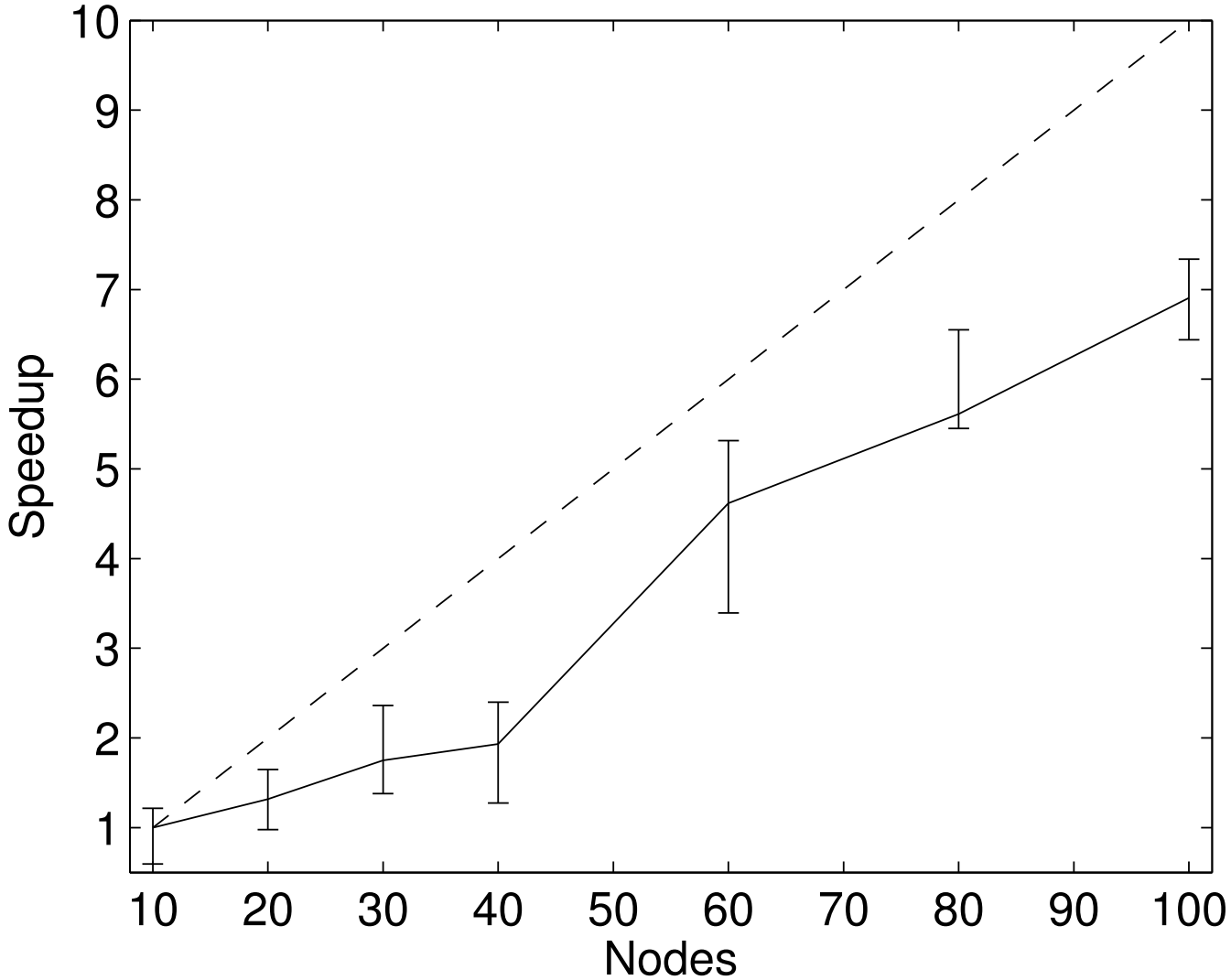
How we do it

- Several tricks play a role, two central to the system
- **Hadoop + AllReduce**
  - AllReduce great to sum gradients, average parameters,...
  - AllReduce doesn't require new job per iteration
  - Hadoop good to schedule problems close to data
  - Hadoop adds faul-tolerance, speculative execution

How we do it

- Several tricks play a role, two central to the system
- **Hadoop + AllReduce**
  - AllReduce great to sum gradients, average parameters,...
  - AllReduce doesn't require new job per iteration
  - Hadoop good to schedule problems close to data
  - Hadoop adds faul-tolerance, speculative execution
- **Hybrid optimization**
  - Initial few passes with online learning+averaging for a reasonable solution
  - Follow by L-BFGS passes with gradient summation for rapid convergence

How we perform



2.1Tfeatures, 17B samples, 16M parameters, 1K nodes, 1-2 hours training time