A Relaxed Synchronization Approach for Solving Parallel Quadratic Programming Problems with Guaranteed Convergence

Kooktae Lee*, Raktim Bhattacharya*, Jyotikrishna Dass**, V N S Prithvi Sakuru**, Rabi N. Mahapatra**

*Aerospace Engineering **Computer Science and Engineering

Texas A&M University

May 24, 2016

Table of Contents

1. Introduction

Large-scale Distributed Optimization Problems with focus on Parallel Quadratic Programming (PQP)

Applications of PQP:

1) least square problems with linear constraints
2) regression analysis and statistics
3) SVMs (Support Vector Machines)
4) lasso (least absolute shrinkage and selection operator)
5) portfolio optimization problems

Problems when implementing parallel computing algorithm

Several critical issues commonly encountered by parallel computing:

- Load imbalance
- Shared memory movement
- Communication overhead
- Synchronization bottleneck

According to the literature [1], the idle process time may be up to 50% of total computation time.

[1] Buchholz, Peter, Markus Fischer, and Peter Kemper. "Distributed steady state analysis using Kronecker algebra." Numerical Solutions of Markov Chains (NSMC'99): 76-95.

Some facts on synchronization:

- Synchronization requires idle process time for multi-core computing devices
- For extreme-scale parallel computing, this leads to waste of computing time
- Need to relax the synchronization penalty!

How to avoid synchronization latency?

1) Asynchronous Computing algorithm:
   - Proceed with computation without waiting values computed by other processors $\Rightarrow$ No need to synchronize data
   - Asynchrony causes randomness in computing values
   - Cannot guarantee the numerical stability and convergence of solution obtained by async. algorithm
2) Relaxed Synchronization approach:
   - Do not synchronize the data and hold synchronization
   - Communication takes places periodically.
   - Objective is to minimize the number of communication (synchronization) $\Rightarrow$ How frequently communicate?

2. Problem Description for Parallel Quadratic Programming (PQP) Problem

The quadratic programming problem considered here is

Quadratic Programming Problem

$$\min_x f(x) \text{ subject to } Ax = b, \tag{1}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$,

$$f(x) := \frac{1}{2}x^T Qx + c^T x,$$

$Q \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite matrix, and $c \in \mathbb{R}^n$.

The cost function $f(x)$ is separable, i.e.,

Assumption

$$f(x) = \sum_{i=1}^{N} f_i(x_i) = \sum_{i=1}^{N} \frac{1}{2} x_i^T Q_i x_i + c_i^T x_i$$

$$Ax = \sum_{i=1}^{N} A_i x_i,$$

where N denotes the total number of subproblems.

Dual Ascent Algorithm

Lagrangian:

$$L(x, y) := f(x) + y^T(Ax - b),$$

where y is the dual variable or the Lagrange multiplier.

$$x_i^{k+1} = \arg\min_{x_i} L_i(x_i, y^k) = -Q_i^{-1}(A_i^T y^k + c), \quad i = 1, \dots, N, \tag{2}$$

$$y^{k+1} = y^k + \alpha^k(Ax^{k+1} - b), \tag{3}$$

where $\alpha^k > 0$ is the step size, the superscript k is the iteration counter, and x_i are partitions of x.

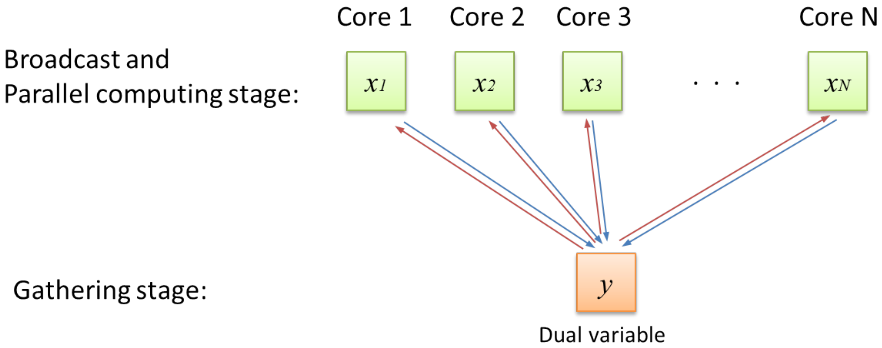


Figure: The schematic of Dual Ascent algorithm

In the gathering stage, the synchronization is necessary and unavoidable!

3. Relaxed Synchronization Approach: Lazy Synchronization

TSDA (Tightly Synchronized Dual Ascent) Algorithm

$$
\begin{aligned}
y^{k+1} &= y^k + \alpha^k (Ax^{k+1} - b) \\
&= y^k + \sum_{i=1}^{N} \alpha_i \left(A_i x_i^{k+1} - \frac{b}{N} \right).
\end{aligned} \tag{4}
$$

LSDA (Lazily Synchronized Dual Ascent) Algorithm

$$
y^{k+1} = y^k + \sum_{i=1}^{N} \alpha_i \left(A_i x_i^{tP+1} - \frac{b}{N} \right), \quad tP \leq k < (t+1)P, \tag{5}
$$

where $t \in \mathbb{N}_0$.

Known: $x_i^{tP+1} = -Q_i^{-1}(A_i^T y^{tP} + c)$

LSDA Algorithm Development

$$y^{k+1} = y^k + \sum_{i=1}^{N} \alpha_i \left(-A_i Q_i^{-1} \left(A_i y^{tP} + c \right) - \frac{b}{N} \right), \quad tP \le k < (t+1)P. \tag{6}$$

when $k = (t+1)P - 1$, we have

$$y^{(t+1)P} = \left(I - P \sum_{i=1}^{N} \alpha_i \left(A_i Q_i^{-1} A_i^T \right) \right) y^{tP} - P \sum_{i=1}^{N} \alpha_i \left(A_i Q_i^{-1} c + \frac{b}{N} \right), \tag{7}$$

where I stands for the identity matrix with a proper dimension.
New dynamics given by:

$$\Rightarrow y^{(t+1)P} = \mathbf{A}(\mathbf{P}) y^{tP} + \mathbf{b}(\mathbf{P}), \quad P \in \mathbb{N}$$

Three issues related to LSDA algorithm

For LSDA algorithm, the following issues have to be resolved.

1) Stability issue:
   Is LSDA algorithm stable?

2) Convergence issue:
   Does LSDA algorithm provide the same solution as compared to TSDA algorithm?

3) Optimality issue:
   What is the optimal synchronization period P^* then?

Lemma

(Stability) The dual variable for LSDA algorithm is stable if and only if

$$\rho\Big(\mathbf{A}(\mathbf{P})\Big) < 1. \tag{8}$$

where $\mathbf{A}(\mathbf{P}) := I - P\sum_{i=1}^{N} \alpha_i \left(A_i Q_i^{-1} A_i^T\right)$ *and the symbol* $\rho(\cdot)$ *denotes the spectral radius of the given matrix (i.e., the largest magnitude of the eigenvalue).*

Proposition

(Convergence) Consider the QP problem that is separable. If the condition (8) holds, then the dual variables y_{LSDA} for LSDA and y_{TSDA} for TSDA converge to the same fixed-point value $y^* := \lim_{k\to\infty} y^k_{TSDA} = \lim_{t\to\infty} y^{tP}_{LSDA}$.

Theorem

(Optimality) For the given parallel QP problem with LSDA technique, the optimal synchronization period $P^\star$ is obtained by

$$P^\star = \max \operatorname*{argmin}_{P\in\mathbb{N}} \max\{|1-\underline{\lambda}(\beta)P|, |1-\bar{\lambda}(\beta)P|\} \tag{9}$$

where $\beta := \sum_{i=1}^{N} \alpha_i A_i Q_i^{-1} A_i^T$, $\underline{\lambda}(\cdot)$ and $\bar{\lambda}(\cdot)$ denote the smallest and the largest eigenvalues of the square matrix, respectively.

4. Experimental Results

Hardware & Software Description

- Hardware:
  40 node cluster of Amazon Web Services (AWS)
  Elastic Cloud Compute (EC2) instances:
  Intel Xeon processors with clock speed up to 3.33 GHz
  One processing unit with 1GB memory

- Software:
  C++ with Armadillo (v5.400.2) – linear algebra library
  MPI framework library (MPICH-v3.1.4) for the inter-node communication

The data was synthetically generated with random values uniformly distributed over $[-1, 1]$. The problem specifics are as follows:

Experimental Setup

1) Number of instances in synthetic dataset, $d = 200,000$.
2) Step size, $\alpha = 0.27$.
3) Optimal Synchronization Period, $P^* = 70$.
4) Stopping threshold, $\epsilon = 10^{-5}$.
5) Cluster Size, $N = \{10, 20, 32, 40\}$.

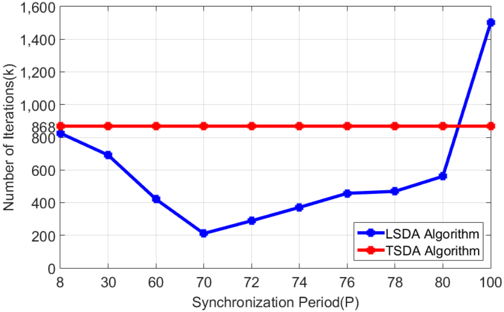


Figure: LSDA algorithm: Number of iterations (k) vs Synchronization Period (P). The number of iterations required for the TSDA algorithm to converge is constant.

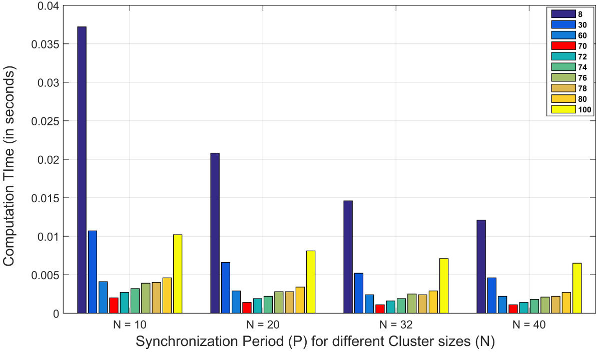


Figure: LSDA algorithm: Computation Time vs Synchronization Period for cluster size N $= \{10, 20, 32, 40\}$

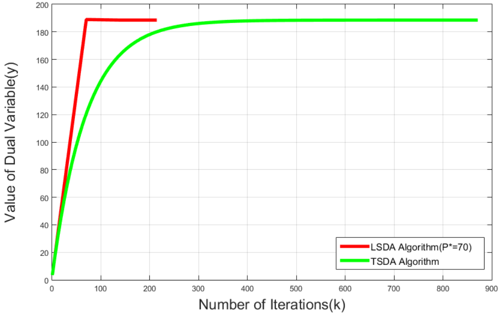
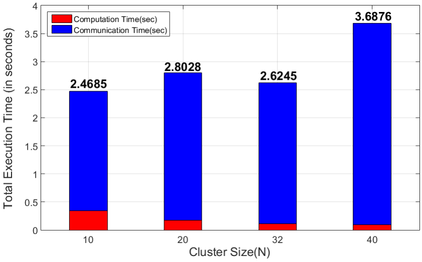


Figure: Dual variable solution vs Number of iterations. LSDA algorithm converges to the optimal solution of the dual variable significantly faster than the TSDA algorithm.



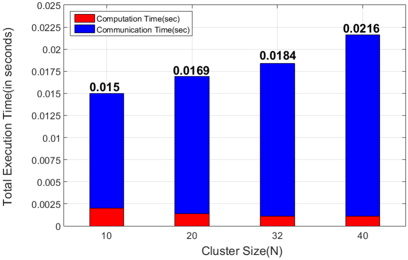


Figure: Total execution time vs Cluster size for TSDA algorithm (left) and for LSDA algorithm (right)

Computing Performance Analysis

Computing Performance Comparison between TSDA & LSDA algorithm

	TSDA algorithm	LSDA algorithm
No. of iteration	868	211
Sync. period	1	70
No. of Sync.	868 (=868/1) times	3 (=211/70) times
Comm. delay reduction	99.65%	
Speedup	160 times	

Summary

1. A relaxed synchronization technique was developed to solve massively parallel large-scale QP problems.
2. Optimal synchronization period is computed analytically with guaranteed convergence.
3. The efficiency of the proposed methods was verified through the real implementation of parallel computing algorithm.

Thank you.

Acknowledgement:

This work was supported by AFOSR grant FA9550-15-1-0071, with Dr. Frederica Darema as the program manager.