Introduction

1. Present a Spark implementation of a custom model
   1. Give many real different customize models (cite some papers)
      1. LDA, LDA-friends
      2. GMM, GMM-friends
      3. BP, BP-friends
2. Show two similar models
   1. ‘small change’ on model
   2. show their respectively Spark Implementation
   3. show their ‘big change’ on Spark Implementation
3. Why not (MLLib) library, but probabilistic programming?
   1. Same example has to show it is NOT changing the instantiation of the existing code (refer to 2b/c examples)
   2. Performance enhancement is neglible compared to the data processing time
   3. Main contribution of code gen is if there’s a (even) small change in model, the change in spark code is significant and can’t be parameterized.
4. Why not SparkR/R/MatLab?
   1. sparkR is just an R interface for spark components.
   2. supported models are limited, current available models are: GMM, k-means, naive bayes and survReg;
   3. cannot generate custom graphical model, the inference code for the custom model needs to be implemented by user in R;
   4. similar to machine learning lib but quite immature.
5. Why not Infer.NET?
   1. We have answer already: not scale (too slow)
6. Why not MLBase?
   1. Frequentists

Our 3(?) contribution:

1. Use of code-gen to solve 3c

(Technical challenge):

1. Graph Property 1: xxx that makes (general) GraphX inefficient

That leads us to devise new (specialized) partitioning schem

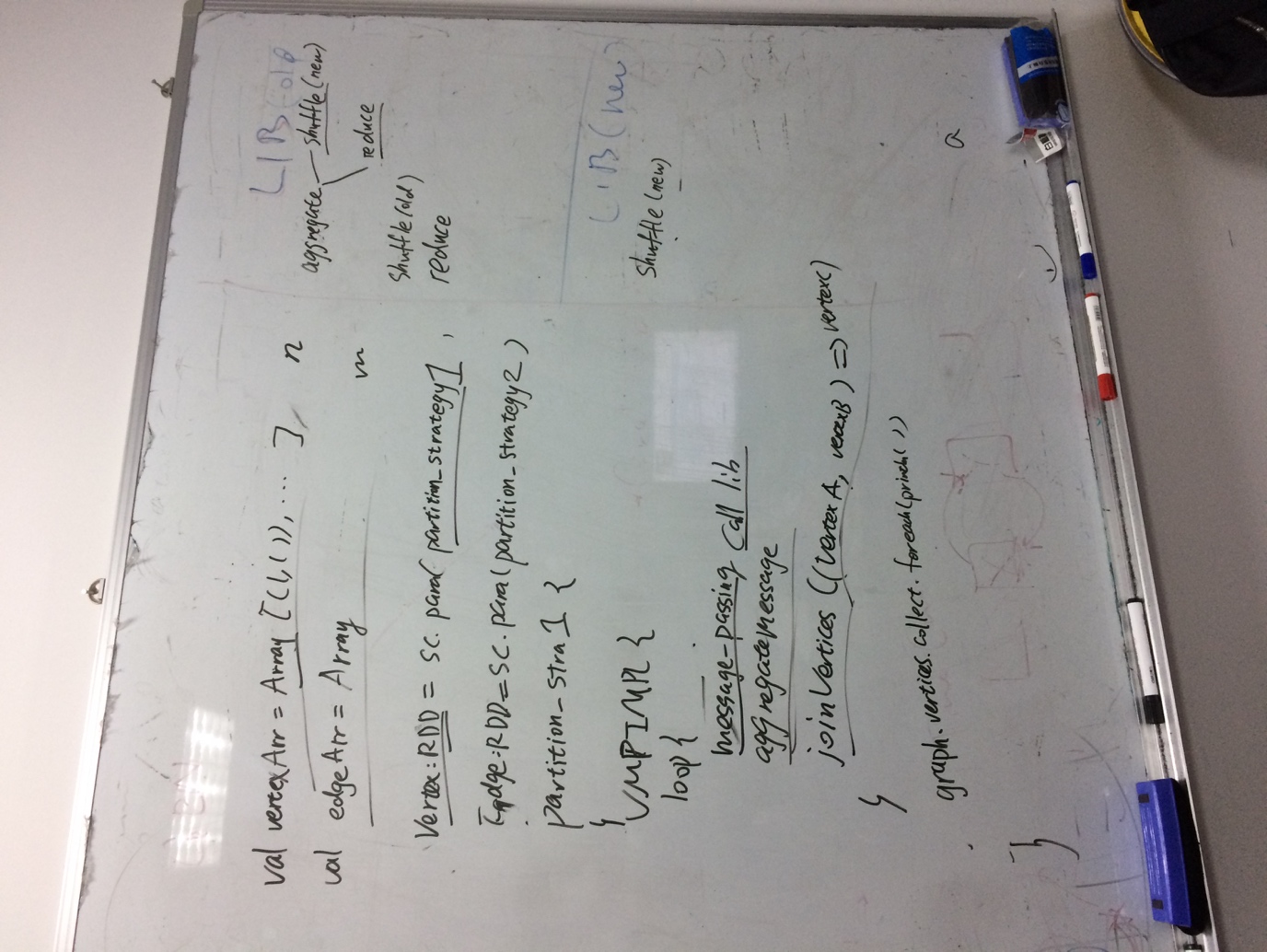
1. Graph Property 2: xxx that makes (general) GraphX inefficient

That leads us to devise new (specialized) partitioning schem

==========================================================

Addition Internal Questions to clarify:

* Why partition-strategy1, 2 can’t be as library (of InferSpark)?
  + If it is not necessary, we write it as library of InferSpark is fine
* What about the generalization of VMP / EP etc?



=========================================================

Related Work

* 1. Distributed Deep Learning Platform
  2. MATLAB, R, SystemML [11], which targets inference code developers instead of end-users.
  3. MLBase, Mahout, MADLib, ML
  4. GraphLab
  5. PyMC
     1. PyMc: Beyesian statistical model in python
     2. can handle custom graphical model;
     3. MC is the only inference algorithm supported;
     4. does not support parrallel computation.
  6. scale-out probabilistic programming has been addressed before, e.g. in the work by Alex Beutel [0][1].

User-view (Formally; no system detail)

* Workflow
  + End-to-end
* Syntax

=======================

4 System-view

[Elsewhere has to mention how different models are being implemented, and how the other ‘future’ unsupported models could be integrated into the system]

* 4.1 Components / Architecture

-

4.2 Compiler / Code Generation

-

* 4.3 InferSpark Runtime Engine