Logistic Regression实证分析

### 主要问题：

|  |  |  |  |
| --- | --- | --- | --- |
| 问题类别 | 问题来源 | 环境, 数据 | 问题描述 |
| 可靠性-OOM | Spark user list-  MLlib-Logistic-Regression-run-out-of-memory-td22210 | 8节点\*(64核，128GB内存)  kddb(19,264,097data, 29,890,095features) | 随着迭代次数的增加会出现OOM |
| 扩展性 | Spark-10078 |  | L-BFGS算法对于特征数非线性扩展，新算法VL-BFGS为特征数无关，可以线性扩展 |
| 扩展性 | MLlib: Scalable Machine Learning on Spark，  Robust Large-Scale Machine Learning in the cloud |  | Logistic Regression with L-BFGS非线性扩展 |

### Issues(<https://issues.apache.org/jira/browse>)

* *Unresolved*

|  |  |  |
| --- | --- | --- |
| 10078 | 问题类别 | 可扩展性与特征数的关系 |
| 版本 | None |
| 问题描述 | L-BFGS算法直接用map-reduce实现需要大量的内存(2md)或大量的map-reduce步骤(2m)，[*d variables and m historical states*] |
| 与什么相关 | 算法 |
| 解决方法 | 新算法Vector-free L-BFGS，避免two loop recursion操作中的点积运算(在driver上还是在executor上？)，提高并行度，内存：O(m2)，map-reduce步骤：3，d可能非常大，新算法d无关。 |
| 解决到什么程度 | 理论[*http://papers.nips.cc/paper/5333-large-scale-l-bfgs-using-mapreduce.pdf*、](http://papers.nips.cc/paper/5333-large-scale-l-bfgs-using-mapreduce.pdf、)  基本的实现[*https://github.com/yanboliang/spark-vlbfgs*](https://github.com/yanboliang/spark-vlbfgs)  需要注意的地方：RDD的特殊性，检查点 |

测试数据集：

*T. Graepel, J.Q. Candela, T. Borchert, and R. Herbrich. Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsofts Bing Search Engine. In Inter-national Conference on Machine Learning, pages 13–20. Citeseer, 2010*

|  |  |  |
| --- | --- | --- |
| 14464 | 问题类别 | 可扩展性与特征数的关系 |
| 版本 | 1.6.0 |
| 问题描述 | 特征数大时，在worker之间传输大阵列(是什么？)存在大量开销，即使数据十分稀疏，数组(阵列？)大小也不变，spark.driver.maxResultSize也要设置非常大来容纳拉回到driver上的数据。 |
| 与什么相关 | 程序 |
| 解决方法 | 有一个使用稀疏向量进行聚合的选项，设置为估计器(estimator?)上的一个参数 |

|  |  |  |
| --- | --- | --- |
| 17471 | 问题类别 | 性能与矩阵压缩存储 |
| 版本 | None |
| 问题描述 | Spark中的矢量具有通过最小化存储需求来选择稀疏或密集表示的压缩方法，矩阵也应该有这种方法，并且包括行存储和列存储。 |
| 与什么相关 | 程序和数据稀疏度 |

|  |  |  |
| --- | --- | --- |
| 17134 | 问题类别 | 性能改进 |
| 版本 |  |
| 问题描述 | 多项Logistic回归使用LogisticAggregator类进行渐变更新，研究重构MLOR以使用2级BLAS操作来进行更新。 |
| 与什么相关 | 算法 |
| 解决方法 | *https://issues.apache.org/jira/browse/SPARK-17134* |

|  |  |  |
| --- | --- | --- |
| 17090 | 问题类别 | OOM |
| 版本 | 2.1.0 |
| 问题描述 | tree aggregate的默认深度可能导致OOM的问题，应该为可设置 |
| 与什么相关 | 程序 |
| 解决到什么程度 | 已解决 |

|  |  |  |
| --- | --- | --- |
| 17201 | 问题类别 | 稳定性 |
| 版本 | None |
| 问题描述 | 没有正则化的softmax回归，Hessian变为奇异，Newton方法可能遇到问题，可能数值不稳定。 |
| 与什么相关 | 程序 |
| 解决方法 | 没有正则化的时候进行正则化 |
| 解决到什么程度 | 是否不稳定待严格测试 |

|  |  |  |
| --- | --- | --- |
| 6567 | 问题类别 | 参数服务器，模型并行 |
| 版本 |  |
| 问题描述 | 每轮迭代中，每个训练点都要对模型进行点积，当模型大到无法放到一台机器中时，需要使用参数服务器，更好的方式是模型并行化。 |
| 与什么相关 | 程序 |
| 解决方法 | 以模型/梯度/边界作为RDD并进行分区 |
| 解决到什么程度 | 实现模型并行化 |

|  |  |  |
| --- | --- | --- |
| 16235 | 问题类别 | 运算错误 |
| 版本 | 1.6.1, 1.6.2, 2.0.0 |
| 问题描述 | mse是真实的4倍 |
| 与什么相关 | 程序 |

* *Resolved*

|  |  |  |
| --- | --- | --- |
| 16008 | 问题类别 | GC |
| 版本 | 2.0.0 |
| 问题描述 | Logistic Regression aggregator序列化不必要的数据(2 \* numFeatures)，将导致性能问题如频繁的GC |
| 与什么相关 | 程序，数据 |
| 解决方法 |  |
| 解决到什么程度 | 解决 |

|  |  |  |
| --- | --- | --- |
| 680 | 问题类别 | 长时间运行 |
| 版本 |  |
| 问题描述 | master重复broadcast 30M数据到100个slaves，集群将挂起 |
| 与什么相关 |  |
| 解决方法 |  |
| 解决到什么程度 | 无法复现 |

## Mailings

|  |  |  |
| --- | --- | --- |
| *http://apache-spark-user-list.1001560.n3.nabble.com/Logistic-Regression-MLLib-Slow-td6996.html* | 问题类别 | 运行时间过长 |
| 版本 |  |
| 问题描述 | 单机(128GB RAM)运行2GB数据，executor使用8核20G内存，用cache需要3.5h，不cache需要5h。 |
| 与什么相关 | 环境设置 |
| 解决方法 | standalone模式运行，并设置driver-memory |

|  |  |  |
| --- | --- | --- |
| *http://apache-spark-user-list.1001560.n3.nabble.com/why-a-machine-learning-application-run-slowly-on-the-spark-cluster-td10910.html#a10921* | 问题类别 | 运行时间过长 |
| 版本 |  |
| 问题描述 | 输入数据:  hdfs存储, 10 part \* 2.3G/part, 1M+ features, sparse, 均匀分布  集群配置：  CentOS, 8 machines, 8 cores, 128G Memory per machine;  executor.memory 64g; spark.default.parallelism 128; spark.akka.timeout 60; storage.memoryFraction 0.7;  kryoserializer.buffer.mb 1024; cores.max 64;  speculation true;  storage.blockManagerTimeoutIntervalMs 60000.  executor不并发启动，一些executor完成了所有的task，而另外一些executor还没有运行task，IO/CPU没有用满，task没有结束时CPU使用率就已经降到1%以下，平均每个stage运行3.5min。 |
| 与什么相关 | 并行度 |
| 解决方法 | repartition(64) |

|  |  |  |
| --- | --- | --- |
| *http://apache-spark-user-list.1001560.n3.nabble.com/MLlib-Logistic-Regression-run-out-of-memory-td22210.html* | 问题类别 | OOM |
| 版本 |  |
| 问题描述 | kddb(# of data: 19,264,097 / 748,401 (testing) # of features: 29,890,095 / 29,890,095 (testing)) , 8 nodes, each with 64 cores, 128GB, 随着迭代次数的增加会出现OOM。 |
| 与什么相关 | 迭代与内存使用的关系 |

该问题在实际运行中也出现过并且可重现，但是社区中没有该问题的解答。需要再做实验并监测内存在每次迭代中的变化，并进行定位。

|  |  |  |
| --- | --- | --- |
| *http://apache-spark-user-list.1001560.n3.nabble.com/MLlib-Logistic-Regression-fails-on-small-data-set-td21666.html* | 问题类别 | OOM |
| 版本 | spark-1.0.2 |
| 问题描述 | 数据：2.1GB libsvm format. feature dimension: 3231961, number of samples: 2156517。  环境：2 machines \* 128GB memory and 64 cores. 120GB for Spark in each node.  内存使用率50%, out-of-heap-memory error |
| 与什么相关 | 环境配置，模型大小？ |
| 解决方法 |  |
| 解决到什么程度 |  |

|  |  |  |
| --- | --- | --- |
| *http://apache-spark-user-list.1001560.n3.nabble.com/mllib-performance-on-cluster-td13290.html* | 问题类别 | 扩展性问题 |
| 版本 | spark-1.0.2 |
| 问题描述 | 集群运行时间是单机运行的3倍 |
| 与什么相关 | 数据大小 |
| 原因 | scheduling tasks, sending t hem out to each worker, and network latencies between the nodes开销大 |
| 解决到什么程度 |  |