Flink/Spark优化技术合作项目任务书

（华为-中科院软件所）

1 背景和场景

当前Spark/Flink等分布式处理框架被广泛应用于大数据处理与分析，然而当出现数据量过大、数据异常、参数配置不当等情况时，应用容易出现运行时错误等可靠性问题。

同时，当我们需要对集群扩容时，大多依据以往的经验来评估。我们需要科学的验证出Flink/Spark的扩展系数以供扩容时参考，如要增加一倍的分布式处理能力需要增加多少机器。

另外，这两年互联网金融的兴起，团伙欺诈（如短信诈骗/淘宝刷钻/信用卡欺诈/不良贷款申请等）的频繁出现，流计算和图算法结合比较适合这类场景，即：Flink, Graph, ML,CEP结合起来。

2 研究目标 1: 大数据框架的可靠性基准测试

2.1 问题需求

学术界研究：

1. 在 MapReduce 应用内存溢出问题方面：Xu [1] 等人通过研究 123 个真实 Hadoop 和 Spark 应用的内存溢出错误，发现了内存溢出错误的三大原因：框架暂存的数据量过大，数据倾斜，及内存使用密集的用户代码。
2. 大数据在线查询分析错误方面：Li [2] 等人研究了 250 个 SCOPE job（运行在微软的 Dryad 框架之上）的故障错误，发现错误主要原因是未定义的列、错误的数据模式、不正确的行格式、内存累计大量中间数据等等。
3. 在大数据系统运行错误方面：Kavulya 等人 [3] 分析了 4100 个运行在 M45 集群上执行失败的 Hadoop jobs。他们发现 36%的故障是数组访问越界错误，还有 23%的故障是 IO 异常。

现实环境中的痛点：

1. 发现在高负载、超大数据集下，Spark 任务在 shuffle 过程中容易出现数据获取失败、任务丢失、IO 异常等问题。另外当小文件过多时，应用在运行中出现一些运行时错误。
2. 某公司在 Spark 上运行机器学习应用时发现在超大数据（100TB）、超高维（上亿维）情况下会出现 driver 运行失败，任务假死等问题。在运行大规模图计算应用时出现内存膨胀、GC 压力大、checkpoint 仍然会进行数据重算等问题。

问题总结：

1. 数据异常
2. 参数配置不当
3. 用户代码缺陷

由于系统缺陷和用户代码缺陷难以预知与检测，我们主要研究数据和参数如何影响应用的运行时错误。

2.2 合作目标

1. 检测运行时错误
   给定应用，快速测试应用是否会出现运行时错误。即应用在什么情况下（数据、配置）会出现运行时错误（OOM、IOException、 Timeout 等）
2. Spark/Flink 可靠性测试
   检测是否存在与运行时错误相关的系统缺陷

2.3 研究方案

自动生成异常数据和参数来组合测试，以便发现应用运行时错误和系统缺陷。这就需要构造一个可靠性测试基准，基准主要包含：

1. Benchmark 应用选择
   - Flink (Sort, SQL)
   - Spark (SQL, ML, Graph)
2. Benchmark 异常数据
   - 基于真实数据，扩展生成异常数据
   - 异常数据特征包含：
     - 数据量大 (>=100GB)
     - 数据分布异常(如 skewed data)
     - 高数据维度(>=百万)
     - 数据稀疏程度高
3. Benchmark 应用配置参数
   系统参数：
   - input split 大小
   - partition 个数
   - buffer 大小等

   应用参数：
   - 迭代次数
   - 聚类个数
   - 决策树深度
   - 分类个数等
4. Benchmark 应用测试方法
   在生成的异常数据上进行参数组合测试

   研究问题：参数组合测试时，参数很多，测试空间太大。如何削减测试空间？
   方法：通过分析参数间独立性，及参数是否与资源使用率、性能相关性来分割参数区间，削减整个参数空间。
5. Benchmark 测试报告
   1. 发现运行时错误，报告详细的错误
   2. 未发现运行时错误
   3. 报告应用在什么数据什么参数上的性能最差或者资源使用最多，及具体的函数关系

2.4 预期成果

1. Benchmark 测试框架代码

2. Flink/Spark 测试报告

2.5 进度安排：

1. 分析理解 Spark/Flink 框架
2. 分析理解 Benchmark 应用
3. 将 Spark Benchmark 上的应用迁移到 Flink 上
4. 设计实现数据生成器
5. 设计实现组合测试方法及空间削减方法
6. 设计实现自动化测试框架
7. 搭建测试环境，部署测试并生成测试报告
8. 分析测试结果

3 研究目标 2：反欺诈场景的实现和调优（研究与改进）

3.1 问题需求

基于关系数据的挖掘和实时计算的研究，反欺诈是Flink图计算的具体场景。

传统的风控和反欺诈依赖于专家系统的规则匹配，在团伙欺诈/淘宝刷钻/金融风控等利用关系网络的场景中规则匹配很难发现这些特征。基于连接关系的图计算系统和图算法在这些场景中能很好的弥补这个问题。

同时，这些欺诈场景一般是在线发生，通过离线计算事后补救效果不大，理想的做法是在线发现这些风险，实时止损。

要研究的具体问题如下：

1. 如何在 Streaming graph 上做在线的图计算
   - 图节点和边不断添加、更改，且与时间相关
   - 算法包括连通图识别、最短路径、K-Core，K-truss, SimRank 等
2. Streaming 数据上做在线机器学习（训练和预测）

3.2 合作目标

1. 能实时识别出团伙欺诈行为（CEP/Graph/Machine Learning 组合应用）
2. CEP 复杂规则匹配出粗粒度欺诈行为（CEP 复杂规则匹配）

3.3 研究方案

基于Flink Streaming引擎和Flink Gelly实现在线图计算

1. 数据准备

   算法设计与实现阶段使用华为公司提供的sample数据和根据真实数据生成的随机数据。算法测试阶段使用华为公司提供的真实数据。

2. 算法实现

   调研Gelly目前的在线图计算情况及要改进的部分，评估能否实现需求，若可则进行算法实现。

3. 算法调优

3.4 预期成果

1. 内核解析文档（Flink Gelly 及 ML）
2. 反欺诈场景的评估和算法实现
3. 在数据集上验证和测试

3.5 进度安排：

1. Flink 图计算框架 / 机器学习内核解析
2. 调研 Flink 图计算能不能实现 graph/ML 的算法
3. 如果算法变成 streaming 形式，那么需要改进哪些部分？
4. 如果实现，实现程度？是否能够达到在线训练 / 预测 / 查询？
5. 算法实现（连通图，最短路径，K-Core，CEP 复杂规则与算法的结合）
6. 算法测试，在 sample 数据和真实数据上测试 CEP 规则/图算法/ML 算法

4 研究目标 3：Flink/Spark 应用扩展性测试

4.1 问题需求

在Flink/Spark集群扩容时，很难科学的根据增加的计算能力确定要扩容的资源。

扩展性是大数据系统的核心特性。然而，用户时常关注的问题是应用是否能够线性扩展。作为系统提供商，为了回答这个问题，我们需要构造一个专门的扩展性测试基准及框架，研究应用扩展性问题。

4.2 合作目标

1. 给定应用（如 SQL、图计算、ML 应用），应用是否可以线性扩展
2. 如果可以线性扩展，扩展系数为多少，那么为了达到某个 SLA，需要的资源量多大（也就是扩展系数是多少）？
3. 如果不能线性扩展，那么瓶颈在哪里？

4.3 研究方案

测试典型应用（如可靠性Benchmark中的应用）的扩展性，分析不能线性扩展的原因。

1. 应用：选用可靠性 Benchmark 中的典型应用
2. 数据：正常数据（采用测试 Spark/Flink 性能的 Benchmark 生成的数据）
3. 扩展性测试方法：给定应用与数据，通过组合参数、变化资源量来测试应用在不同规模资源下的性能与资源利用率
4. 扩展性分析方法：通过分析运行信息与资源规模的函数关系来判断是否具有扩展性，进一步细粒度分析各个执行阶段及代码，定位达不到线性扩展的瓶颈
5. 给出测试报告与瓶颈分析报告

4.4 预期成果

1. 扩展性测试框架
2. 测试报告

4.5 进度安排：

1. 分析理解 Spark/Flink 框架
2. 分析理解 Benchmark 应用
3. 设计实现数据生成器、组合测试框架
4. 设计实现应用性能监控系统
5. 搭建测试环境，部署测试并生成测试报告
6. 分析瓶颈原因

5 需求规格

三个研究目标的需求规格如下：

关键需求	需求描述	优先级
大数据框架的可靠性基准测试	Flink/Spark在各种系统参数配置和各种测试数据下，组合后测试框架的可靠性，即运行时错误和系统缺陷	高
反欺诈场景的实现和调优（研究与改进）	结合Flink CEP、Gelly、ML等技术，实现反欺诈场景中发掘出团伙欺诈行为。	高
Flink/Spark应用扩展性测试	测试出Flink/Spark是否可线性扩展，以及瓶颈在哪里	中

6 验收场景

SN	验收场景	验收场景说明
	静态场景	在华为提供的服务器上运行待验收的代码，跑出测试结果

7 交付件

7.1 软件交付件

SN	软件交付件名称	数量	备注
1	基准Benchmark框架代码和测试用例	1	实现要求的Benchmark功能的项目代码和测试用例
2	反欺诈场景的功能实现	1	代码基于Flink框架

3	扩展性测试框架代码	1	扩展性测试框架以及测试报告

7.2 非软件交付件

SN	非软件交付件名称	数量	备注
1	Flink/Spark基准Benchmark调研报告	1	调研研究目标1中的Benchmark报告，列出Flink/Spark中存在的缺陷
2	Flink Gelly/ML可行性报告	1	Gelly/ML能否满足发欺诈场景，如果不能满足，要列出问题所在以及改进的方式
3	Flink Gelly/ML的内核解析文档	1	画图和文字两种形式结合，说明内核内在逻辑结构
4	Flink/Spark扩展性报告	1	介绍Flink/Spark的扩展系数以及生成该系数的内在逻辑

8 项目里程碑项目完成时间

共计9个月时间

阶段	阶段说明	阶段完成时间	主要工作内容	输出件	应达到的标准
1	调研阶段	T+3	Flink Benchmark的可能性；Flink在线图计算的可行性；扩展系数的可行性	调研报告1份，	形成可能的项目解决方案
2	实现阶段	T+10	三个合作目标的实现	代码实现以及产生报告	能够部署切实可行的测试平台
3	优化阶段	T+12	对三个合作目标进一步优化	代码实现以及产生报告	能够部署切实可行的测试平台

9 验收标准

在上述指定验收场景下，验收标准如下：

验收项	验收标准	备注
1. 软件	符合5中的需求规格要求；可在约定的系统环境上实现相应功能和性能的复现。代码	代码需通过华为方验收小组检视

	简洁及较好的可读性，测试结果和代码需要通过华为方验收小组评审。	
2. 文档	调研报告：材料全面清晰，结论鲜明； 规格、设计说明书等：体现目标及设计思想，指导理解代码，与代码实现一致，能够支撑软件规格达成预定的目标。	需通过华为方验收小组评审
3. 测试报告	测试结果真实可信，具有可验证性	需通过华为方验收小组方评审

10 参考文献

[1] L. Xu, W. Dou, F. Zhu, C. Gao, J. Liu, H. Zhong, J. Wei. A Characteristic Study on Out of Memory Errors in Distributed Data-Parallel Applications. In the 26th IEEE International Symposium on Software Reliability Engineering (*ISSRE* 2015).

[2] S. Li, H. Zhou, H. Lin, T. Xiao, H. Lin, W. Lin, and T. Xie, "A characteristic study on failures of production distributed data-parallel programs," in 35th International Conference on Software Engineering (*ICSE*), 2013, pp. 963–972.

[3] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan, "Analysis of traces from a production mapreduce cluster," in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (*CCGrid*), 2010, pp. 94–103.