Parallel *k*-means on stream data

1st Author

1st author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

1st author's email address

2nd Author

2nd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

2nd E-mail

3rd Author

3rd author's affiliation  
1st line of address  
2nd line of address  
Telephone number, incl. country code

3rd E-mail

**Abstract**

CluStream、streamKM++、mini-batch k-means等k-means算法的变种可用于流式环境下，本文分析了这些算法的特征，包括算法的定义、在计算、参数更新和反馈上的特点，总结了算法的并行化情况，提出算法的评价方法，对测试结果进行评价，对算法的设计人员提供算法在流式化和并行化设计上的选择和评价的指导。

# Introduction

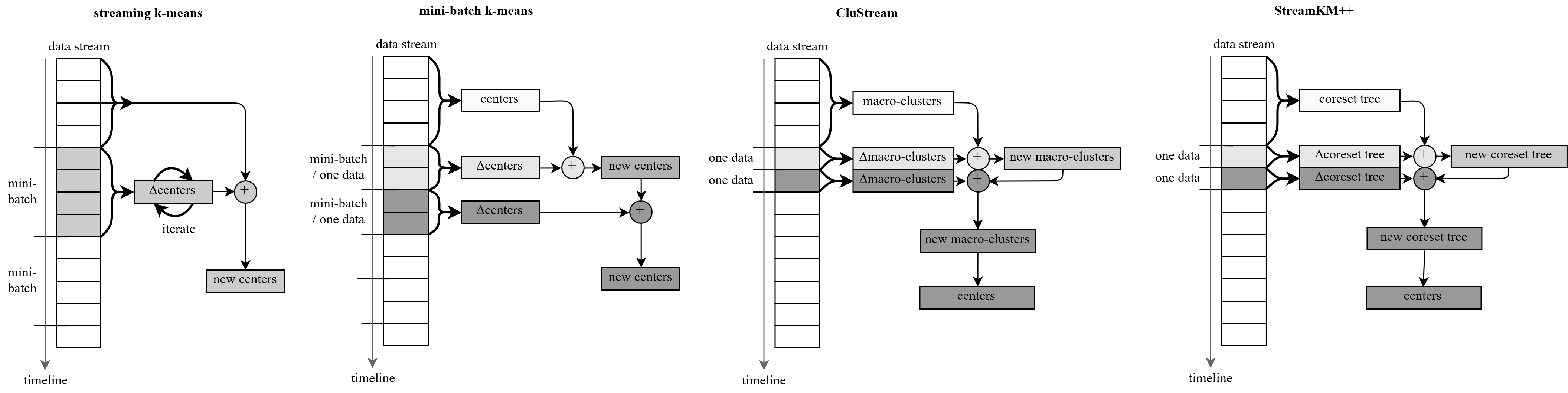
k-means算法是最常用的聚类算法之一，广泛应用与文档分类、犯罪地点识别、客户分类、欺诈检测等场景，这些场景中数据往往是流式且非平稳分布的，因此模型需要随着时间变化。

流式数据上，传统批处理的k-means随着训练数据的增多重复训练全部历史数据，导致算法的实时性差。

流式数据上的k-means聚类算法的经典变种包括CluStream、streamKM++、mini-batch k-means等，他们将历史的数据用历史信息参数的形式表达。目前这些算法在StreamDM、SAMOA、Spark Streaming、Angel等批处理和流处理机器学习平台上实现。

那么他们分别有哪些优缺点，当前并行化情况如何，算法设计人员如何设计对应算法的并行化方式？

为了帮助算法设计人员和使用人员能够根据数和应用场景需求选择合适的流式k-means算法，本文总结了算法的特征、并行化方法、对算法进行测试和评价，回答的研究问题包括：



Figure

1. 算法分别是什么计算模式？
2. 算法的并行化情况和并行化方法如何？
3. 如何对流式k-means算法进行测试，测试指标包括哪些，如何测试，以及如何评价？

论文贡献：

1. 总结了算法特征、计算模式。
2. 基于算法的现有实现，在spark streaming上并行化实现并改进。
3. 根据算法的特征进行针对性测试，提出算法的评价指标和评价方法。
4. 指导算法设计人员如何根据算法特征、数据特征和场景需求并行化设计算法。

# Background

## 流数据上的k-means算法

Table 1

|  |  |  |  |
| --- | --- | --- | --- |
| 算法 | 模型表示 | 计算模式 | 发布时间 |
| Clustream[3] | 微簇 | Online + sketch | 2003 |
| streamKM++[2] |  | Online + sketch | 2012 |
| Mini-bath k-means[1]/ Spark Streaming k-means | 聚类中心(centers) | Online | 2010 / 2015 |

算法在计算模式上的对比如figure 1所示。

【其他streaming k-means相关文献】

Zhang Y, Tangwongsan K, Tirthapura S. Streaming algorithms for k-means clustering with fast queries[J]. arXiv preprint arXiv:1701.03826, 2017.

Braverman V, Meyerson A, Ostrovsky R, et al. Streaming k-means on well-clusterable data[C]//Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2011: 26-40.

Ailon N, Jaiswal R, Monteleoni C. Streaming k-means approximation[C]//Advances in neural information processing systems. 2009: 10-18.

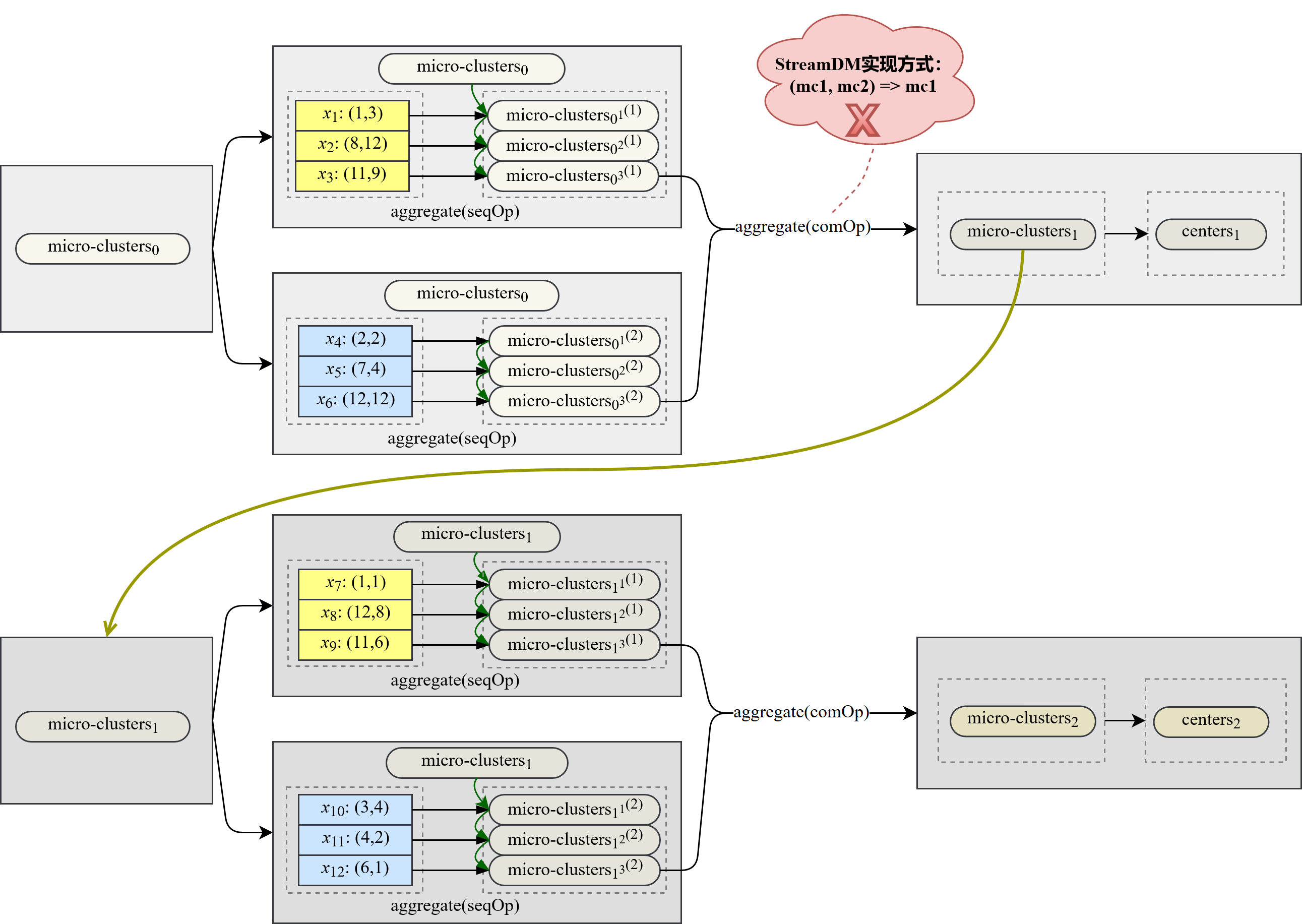


Figure CluStream分布式实现

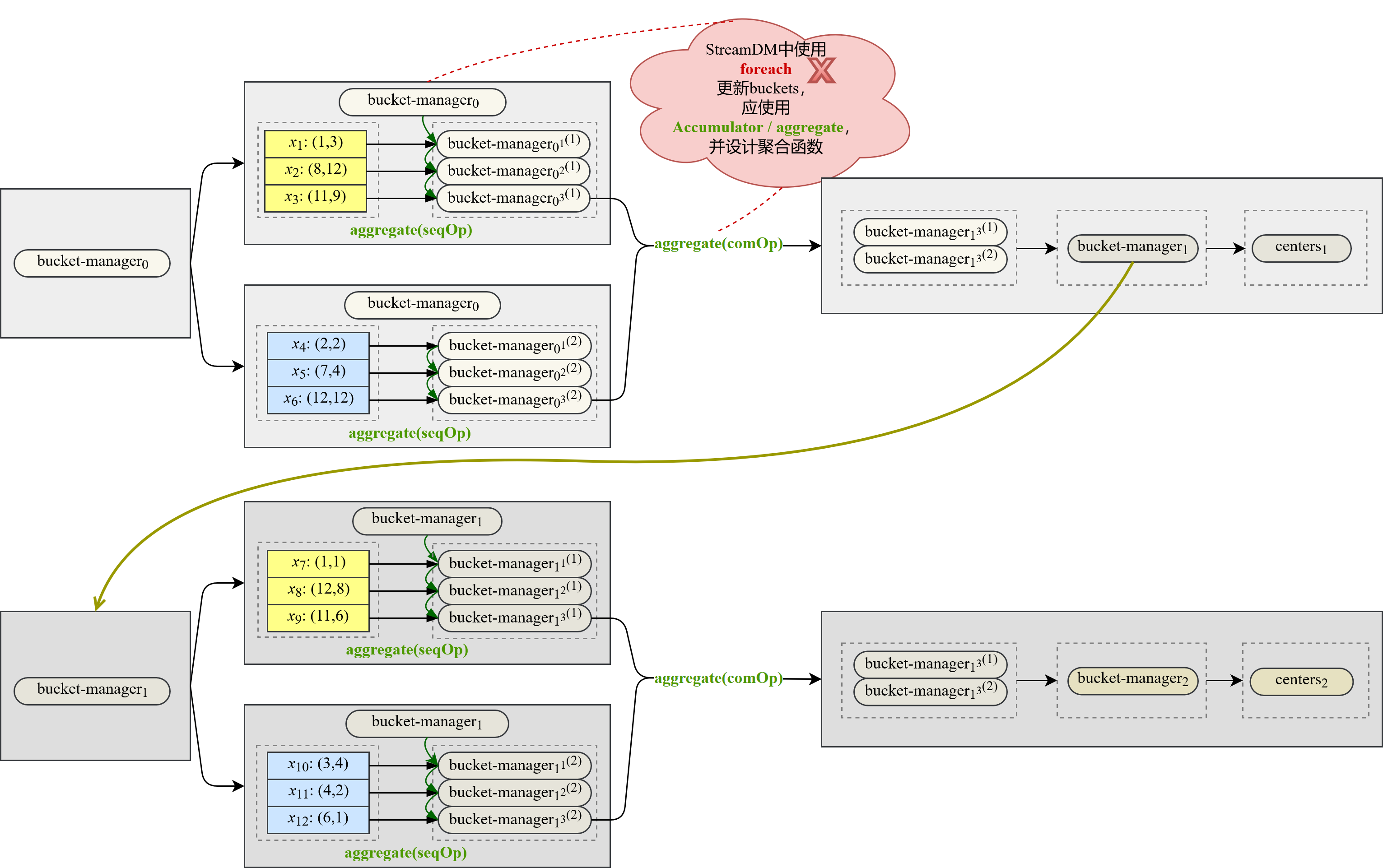


Figure StreamKM++分布式实现

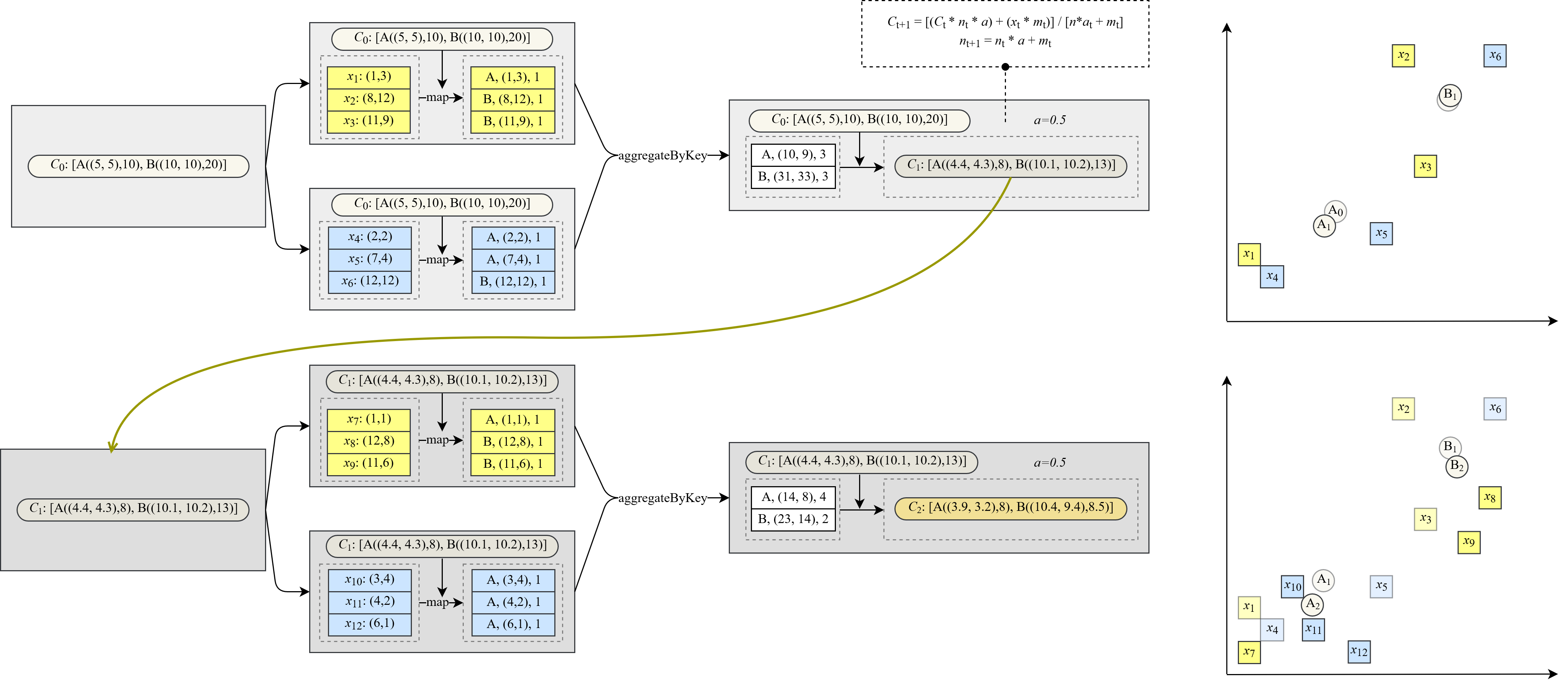


Figure Spark Streaming KMeans分布式实现

## 并行化实现现状

|  |  |  |
| --- | --- | --- |
| 算法 | 并行方式 | 实现平台 |
| CluStream | 数据并行 | StreamDM |
| StreamKM++ | 数据并行 | StreamDM |
| Mini-batch k-means | 数据并行 | Spark Streaming |
| 数据并行+模型并行 | Angel |

当前并行化实现主要几种在StreamDM平台、Spark Streaming和Angel上，其中StreamDM和Angel都是基于Spark实现的。Mini-batch k-means在Spark Streaming上的并行化流式实现称为Spark streaming k-means。将现有的并行化实现CluStream、StreamKM++和Spark streaming k-means数据流图表达为Figure 2、Figure 3、Figure 4所示。StreamDM中实现的CluStream、StreamKM++算法在参数更新和聚合操作上存在错误，其中CluStream算法中没有设计参数的聚合操作，StreamKM++使用foreach算子**[见附录]**更新参数。应该用全局的Accumulator或aggregate算法改写并设计聚合函数。

更改代码，包括CluStream的aggregate(comOp)，实现原理为每两个微簇集合合并时找到最近的q个微簇；StreamKM++的aggregate，实现原理为aggregate(seqOp)对每个数据更新bucketmanager，aggregate(comOp)结合两个bucketmanager时，用其中一个bucketmanager中的每个元素，更新另一个bucketmanager。

实现的算法时间复杂度如下。

CluStream: 一个微批

aggregate(seqOp)

aggregate(comOp) (*p*: #partition)

micro-clusters - cluster

StreamKM++:

aggregate(seqOp) (*m*: coreset size)

aggregate(comOp)

bucketmanager - cluster

Streaming *k*-means:

1. 优化

合并操作的区别：clustream是两两归并，streamKM++是fold操作。

CluStream优化用treeAggregate进行层次归并。

1. 空间复杂度分析

Streaming k-means:

StreamKM++:

CluStream:

1. 可变参数总结
2. 数据参数

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 影响 |
| d | 维度 | 见时间复杂度 |
| n | Batch大小 | 实时性 |
| Rate | 流速 | 流处理速度 |
| Sparsity | 稀疏程度 | 计算最近中心时复杂度与稀疏程度关系分析 |

1. 算法超参

CluStream

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 取值 |
| k | 聚类中心个数（模型维度） | 依照数据 |
| q | 微簇个数 | 远大于k，与单机内存相关 |
| initial\_buffer\_size | 初始化数据集大小 | 1000（默认） |
| max\_iterations | 最大迭代次数 | 100（默认） |
| Radius | 微簇允许加入新数据的半径阈值 | 2（默认） |
| M | 微簇时间戳标准差，从而进行过时分析 | 100（默认） |

StreamKM++

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 取值 |
| k | 聚类中心个数（模型维度） | 依照数据 |
| S | Coreset大小 |  |
| Window\_size | Window大小 |  |
| max\_iterations | 最大迭代次数 | 1000（默认） |

Streaming k-means

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 取值 |
| k | 聚类中心个数（模型维度） | 依照数据 |
| Decay\_factor | 衰减系数 | 0.5（默认） |
| Weight | 初始化各个簇的权重 | 100（默认） |
| max\_iterations | 最大迭代次数 | 1000（默认） |

# Methodology

## 特征分析

计算步骤：

CluStream: 1. 计算新到数据导致的micro-clusters增量；2. 更新micro-clusters；3.根据当前micro-clusters计算最新聚类中心。

StreamKM++: 1. 计算新到数据导致的bucket-manager增量；2.更新bucket-manager；3.根据当前bucket-manager计算最新聚类中心。

Streaming k-means：1. 计算新到数据导致的聚类中心和其他统计信息（计数）的增量；2. 更新聚类中心和统计信息。3. 得到聚类中心。

将计算步骤概括为：

参数增量计算；参数更新；模型计算

在流式化和并行化上的相同点和不同点：参数增量聚合上不同；

## 并行化

改进的点：参数增量的聚合

# Experimental results

## Experimental setup（实验环境）

We performed the evaluation on 9 *mn4.2xlarge* nodes on Alibaba Cloud. One node serves as the master node, and the others are worker nodes. Each node has 4 physical cores (8 virtual cores), 32GB RAM, and 1200GB disk with 3000 IOPS.

## 测试方案

测试数据集：KDD1999

生成方法：固定速率+对应时间戳？

训练数据和测试数据划分：数据前90%为训练数据，后10%为测试数据。

指标：

**流处理速度：**对照三个算法处理同样时间内生成的数据的时间，A方案：横坐标为输入数据量，纵坐标为处理这些数据的处理时间，画曲线（需要改程序，给每条数据到sink端打上时间戳？）；~~B 方案：横坐标为数据生成的速率，测扩展性~~

**计算精度（SSE/SSQ）：**如何计算。如何测试，测试的数据时间和模型时间之间的关系。相同数据的对比。

测试的影响因素为算法共有的参数：

横坐标为窗口大小（同步频率），纵坐标为最终获得的模型的SSQ（测试数据为后10%，训练数据为前90%，最终获得的模型指90%的数据处理结束后）

**扩展性：**相同数据在相同流速下不同并行度的处理时间。

可视化：中心点可视化，在二维数据上测试。（单机？）

## results

**StreamingKMeans流处理速率测试**

# Discussion

# Related work

**并行化做了哪些工作，对比streamDM有哪些改进.目前做到什么。**

# Conclusions

# REFERENCES

1. Sculley D. Web-scale k-means clustering[C]//Proceedings of the 19th international conference on World wide web. ACM, 2010: 1177-1178.
2. Ackermann M R, Märtens M, Raupach C, et al. StreamKM++: A clustering algorithm for data streams[J]. Journal of Experimental Algorithmics (JEA), 2012, 17: 2.4.
3. Aggarwal C C, Philip S Y, Han J, et al. -A Framework for Clustering Evolving Data Stream- s[C]//Proceedings 2003

VLDB Conference. 2003: 81-92

附录

图1 StreamKM++算法在StreamDM中实现的更新函数

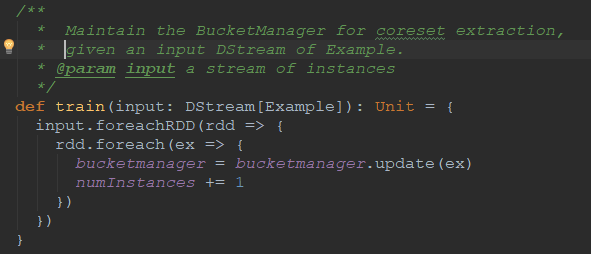
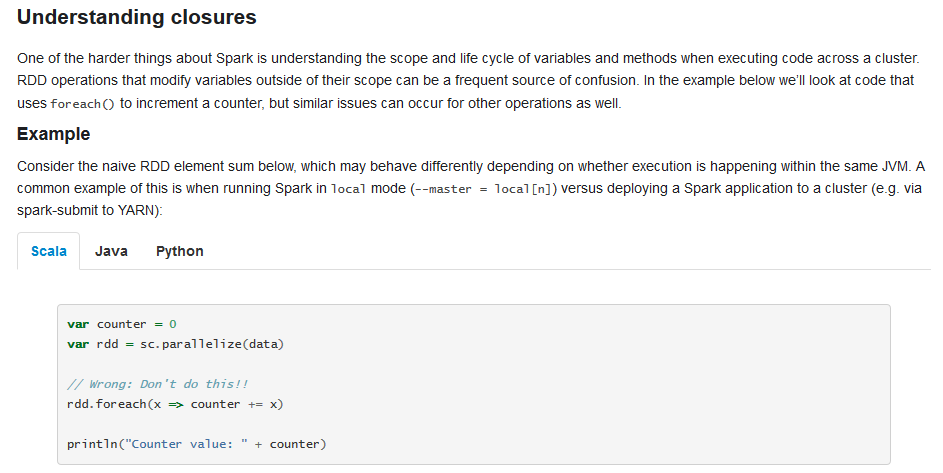


图2 foreach算子使用闭包结构不应该被用来改变全局状态

（来源：[https://spark.apache.org/docs/latest/rdd-programming-guide.html#rdd-operations](https://spark.apache.org/docs/latest/rdd-programming-guide.html" \l "rdd-operations)）



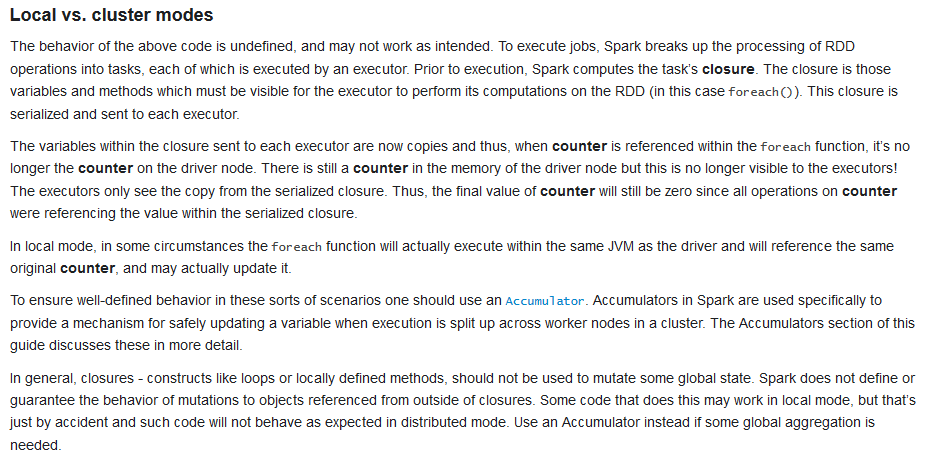


图3 CluStream分布式实现

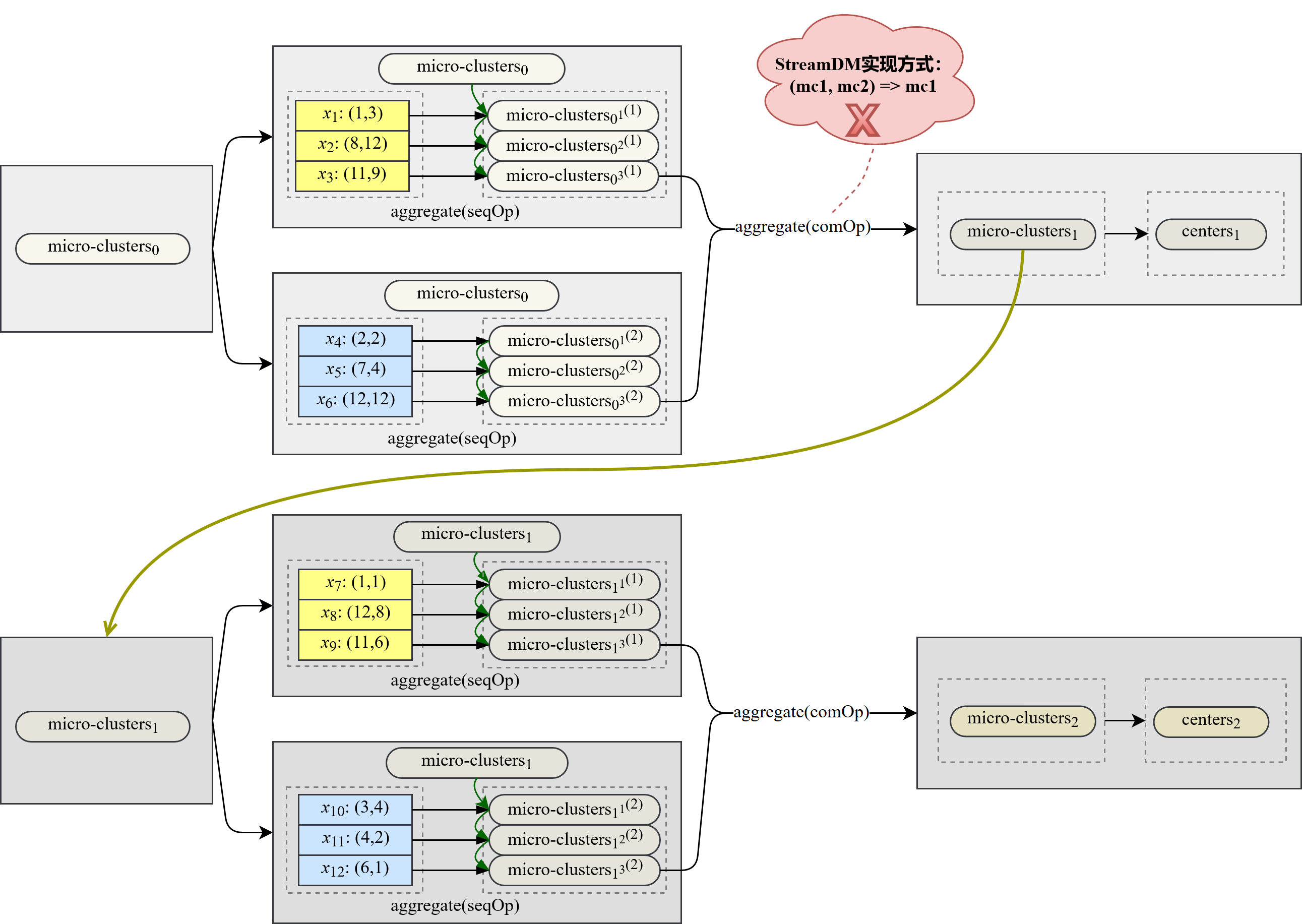


图4 StreamKM++分布式实现

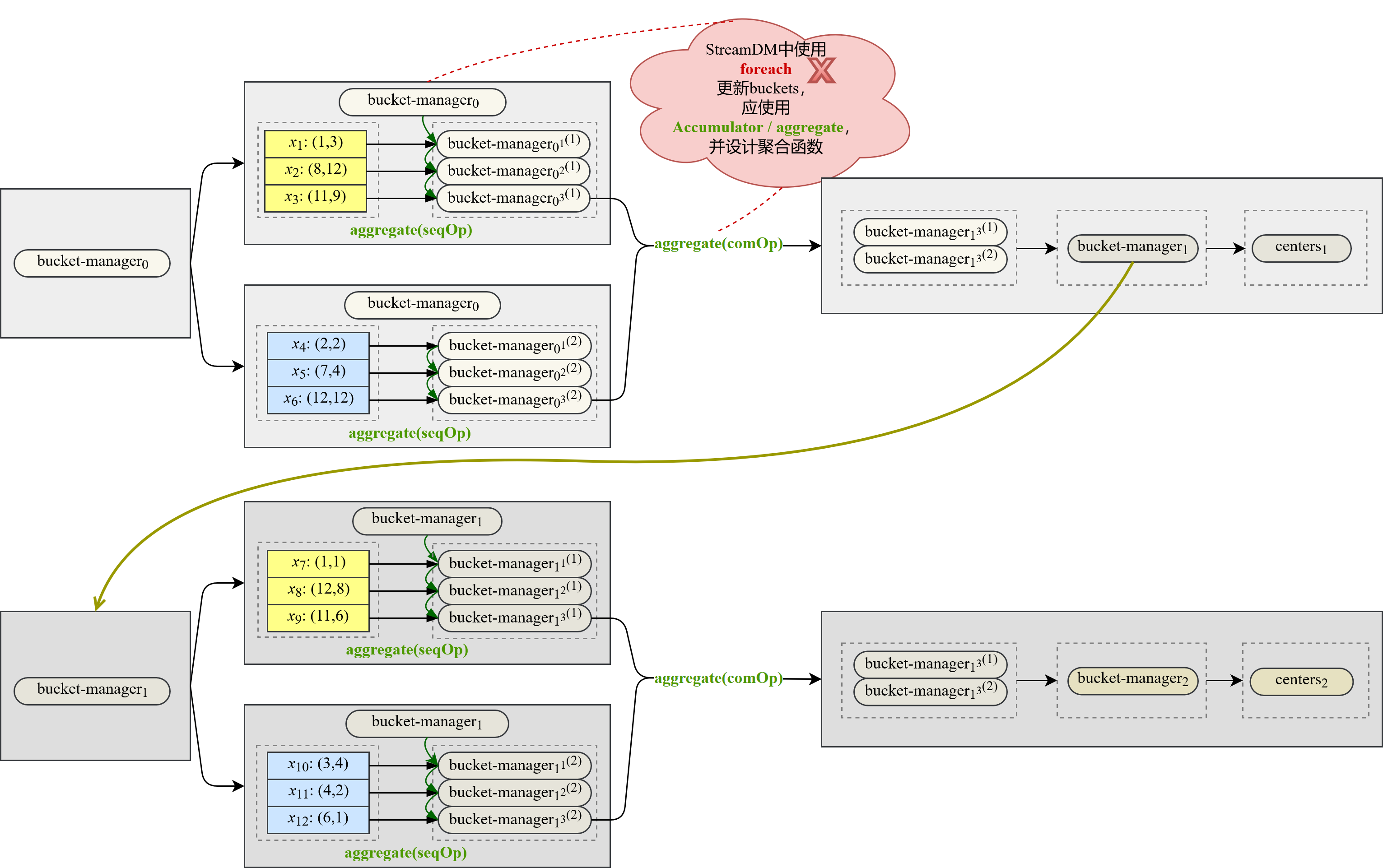


图5 Spark Streaming KMeans分布式实现

