【技术帖】Apache Kylin 高级设置：聚合组（Aggregation Group）原理解析

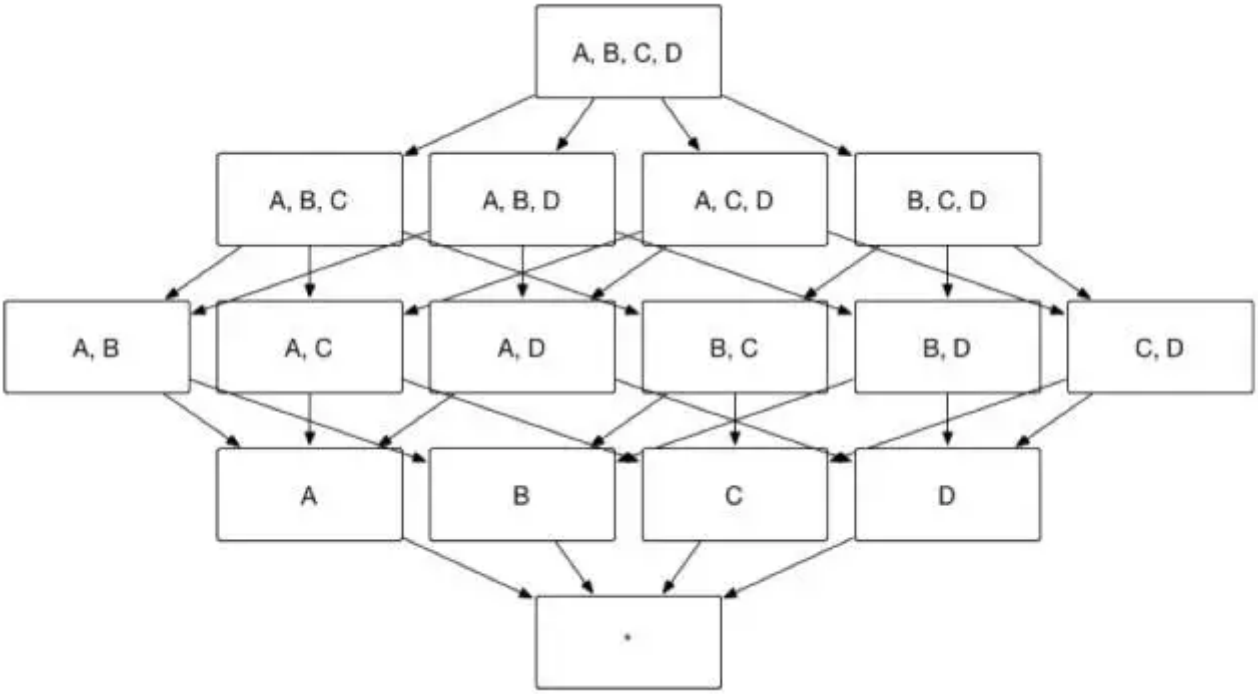
原创 2017-03-31 施继成 apachekylin

点击上方蓝色 apachekylin 可以关注我哟

> 随着维度数目的增加，Cuboid 的数量会爆炸式地增长。为了缓解 Cube 的构建压力，Apache Kylin 引入了一系列的高级设置，帮助用户筛选出真正需要的 Cuboid。这些高级设置包括聚合组（Aggregation Group）、联合维度（Joint Dimension）、层级维度（Hierachy Dimension）和必要维度（Mandatory Dimension）等。
>
> **本文将着重介绍聚合组的实现原理与应用场景实例。**今后还会有相关系列技术文档发布，敬请期待。

作者 | 施继成 翟鹿渊
编辑 | Zoe

众所周知，Apache Kylin 的主要工作就是为源数据构建 N 个维度的 Cube，实现聚合的预计算。理论上而言，构建 N 个维度的 Cube 会生成 2^N 个 Cuboid， 如图 1 所示，构建一个 4 个维度（A，B，C, D）的 Cube，需要生成 16 个Cuboid。



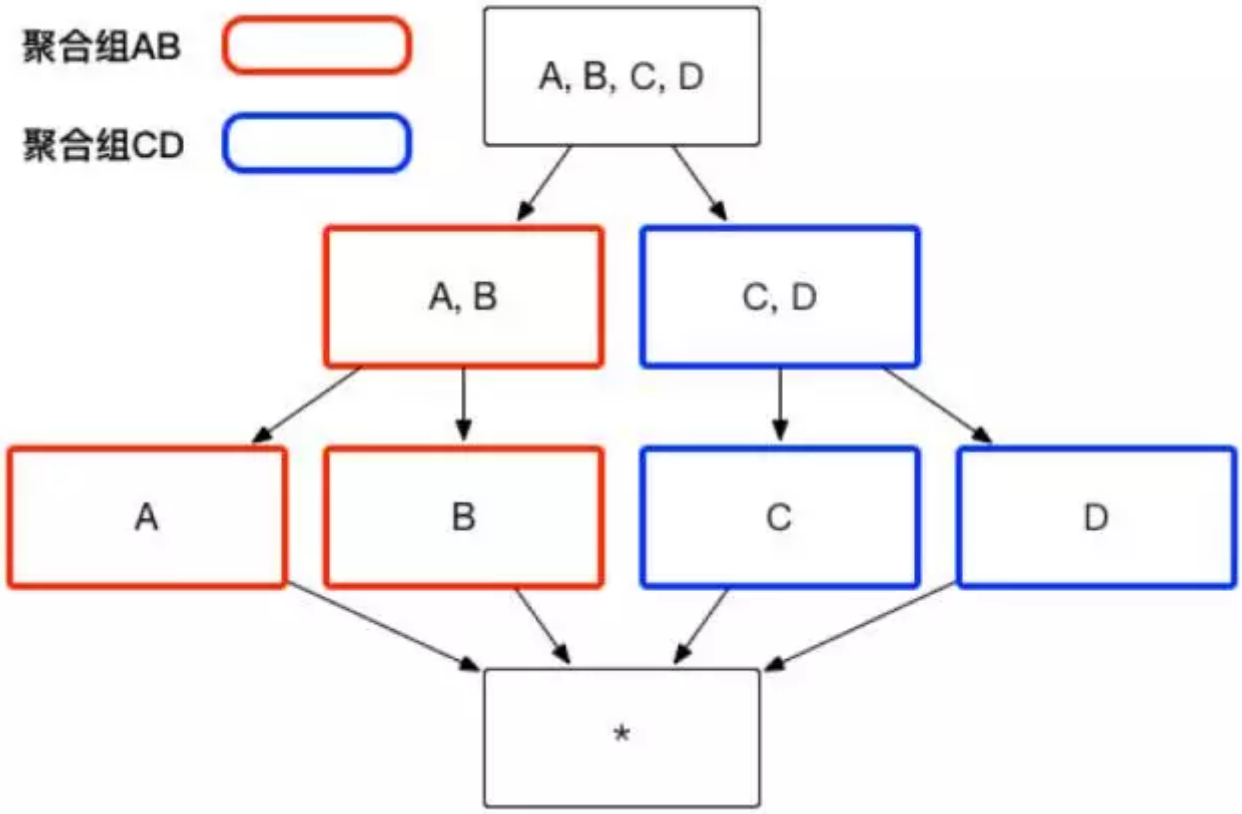
（图1）

随着维度数目的增加 Cuboid 的数量会爆炸式地增长，不仅占用大量的存储空间还会延长 Cube 的构建时间。为了缓解 Cube 的构建压力，减少生成的 Cuboid 数目，Apache Kylin 引入了一系列的高级设置，帮助用户筛选出真正需要的 Cuboid。这些高级设置包括聚合组（Aggregation Group）、联合维度（Joint Dimension）、层级维度（Hierachy Dimension）和必要维度（Mandatory Dimension）等，本系列将深入讲解这些高级设置的含义及其适用的场景。

本文将着重介绍聚合组的实现原理与应用场景实例。

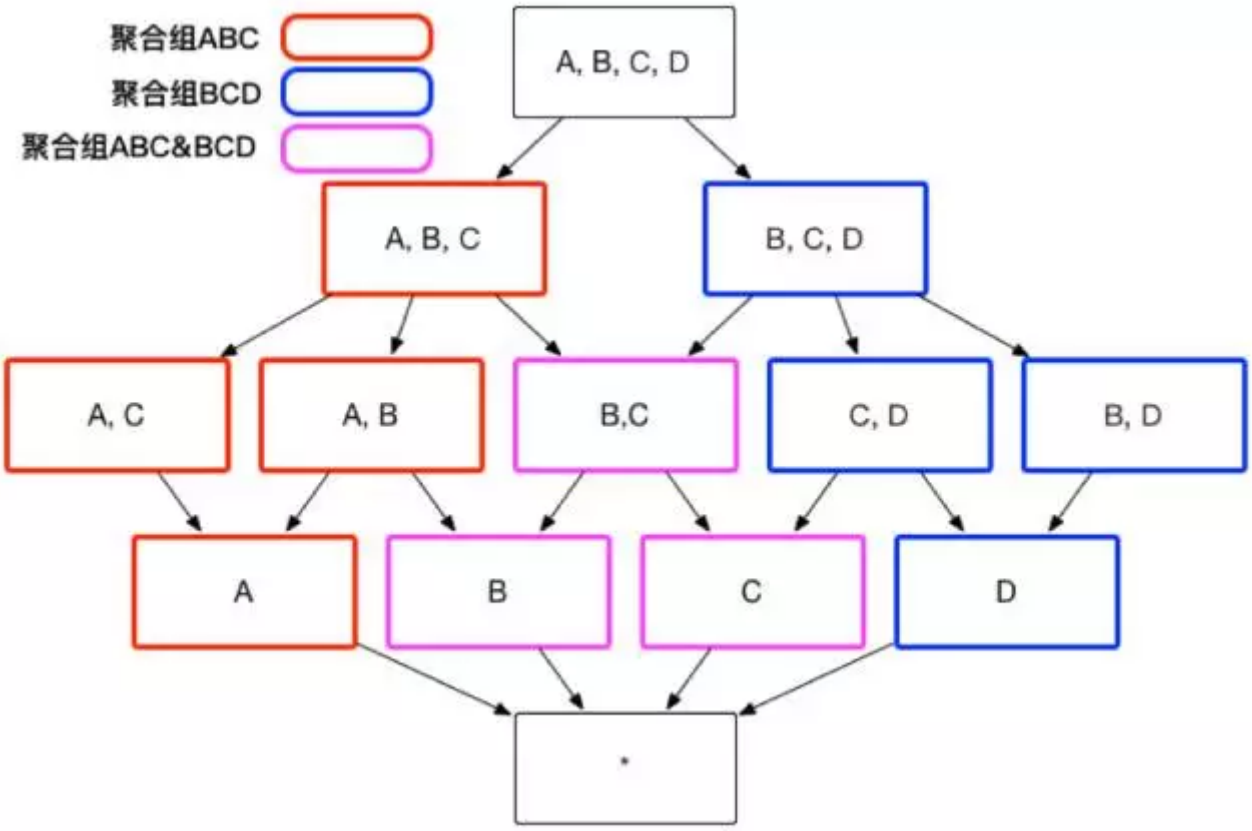
聚合组（Aggregation Group）

用户根据自己关注的维度组合，可以划分出自己关注的组合大类，这些大类在 Apache Kylin 里面被称为聚合组。例如图 1 中展示的 Cube，如果用户仅仅关注维度 AB 组合和维度 CD 组合，那么该 Cube 则可以被分化成两个聚合组，分别是聚合组 AB 和聚合组 CD。如图 2 所示，生成的 Cuboid 数目从 16 个缩减成了 8 个。



（图2）

用户关心的聚合组之间可能包含相同的维度，例如聚合组 ABC 和聚合组 BCD 都包含维度 B 和维度 C。这些聚合组之间会衍生出相同的 Cuboid，例如聚合组 ABC 会产生 Cuboid BC，聚合组 BCD 也会产生 Cuboid BC。这些 Cuboid不会被重复生成，一份 Cuboid 为这些聚合组所共有，如图 3 所示。

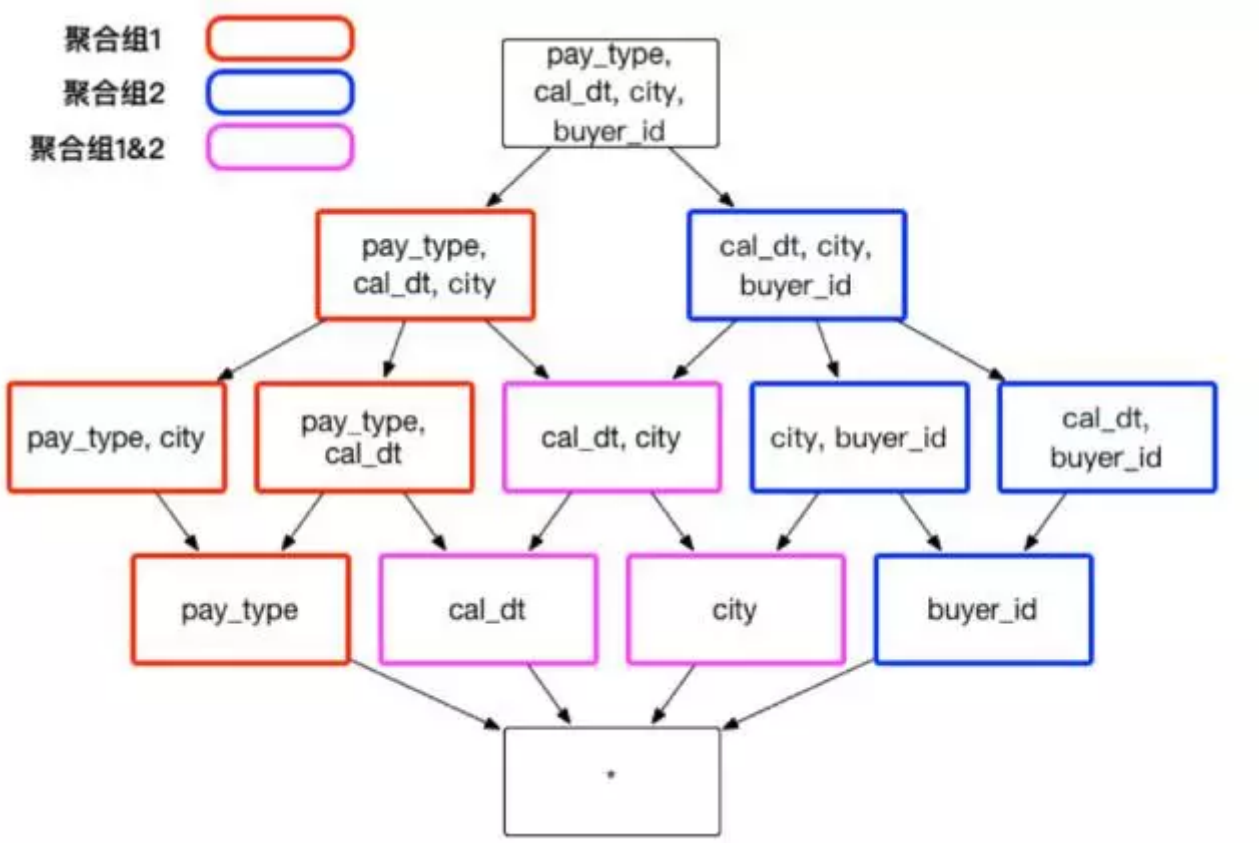


（图3）

有了聚合组用户就可以**粗粒度**地对 Cuboid 进行筛选，获取自己想要的维度组合。

应用实例

假设创建一个交易数据的 Cube，它包含了以下一些维度：顾客 ID buyer_id 交易日期 cal_dt、付款的方式 pay_type 和买家所在的城市 city。有时候，分析师需要通过分组聚合 city 、cal_dt 和 pay_type 来获知不同消费方式在不同城市的应用情况；有时候，分析师需要通过聚合 city 、cal_dt 和 buyer_id，来查看顾客在不同城市的消费行为。在上述的实例中，推荐建立两个聚合组，包含的维度和方式如图 4 ：



（图4）

聚合组 1： [cal_dt, city, pay_type]

聚合组 2： [cal_dt, city, buyer_id]

在不考虑其他干扰因素的情况下，这样的聚合组将节省不必要的 3 个 Cuboid: [pay_type, buyer_id]、[city, pay_type, buyer_id] 和 [cal_dt, pay_type, buyer_id] 等，节省了存储资源和构建的执行时间。

Case 1:

SELECT cal_dt, city, pay_type, count(*) FROM table GROUP BY cal_dt, city, pay_type 则将从 Cuboid [cal_dt, city, pay_type] 中获取数据。

Case2:

SELECT cal_dt, city, buy_id, count(*) FROM table GROUP BY cal_dt, city, buyer_id 则将从 Cuboid [cal_dt, city, pay_type] 中获取数据。

Case3 如果有一条不常用的查询:

SELECT pay_type, buyer_id, count(*) FROM table GROUP BY pay_type, buyer_id 则没有现成的完全匹配的 Cuboid。

此时，Apache Kylin 会通过在线计算的方式，从现有的 Cuboid 中计算出最终结果。

小结

Apache Kylin 作为一种多维分析工具，其采用预计算的方法，利用空间换取时间，提高查询效率。本文介绍了 Apache Kylin 的高级设置中聚合组的部分，聚合组适用于当分析师粗粒度地关注某些维度去进行分组聚合的场景。之后的文章我们还将就 Apache Kylin 其他的高级设置的使用方法和使用场景做详细介绍，敬请期待。

您可能还会想看

【技术贴】如何部署Apache Kylin集群实现负载均衡？
【技术帖】Apache Kylin v2.0.0 Beta尝鲜版上线！！！
【福利帖】《Apache Kylin权威指南》正式发售
【技术贴】揭秘Apache Kylin V2.0新特性：字典编码模块的优化
【案例分享】Apache Kylin在美团点评的应用

长按图片识别二维码关注
Apache Kylin官方公众号