

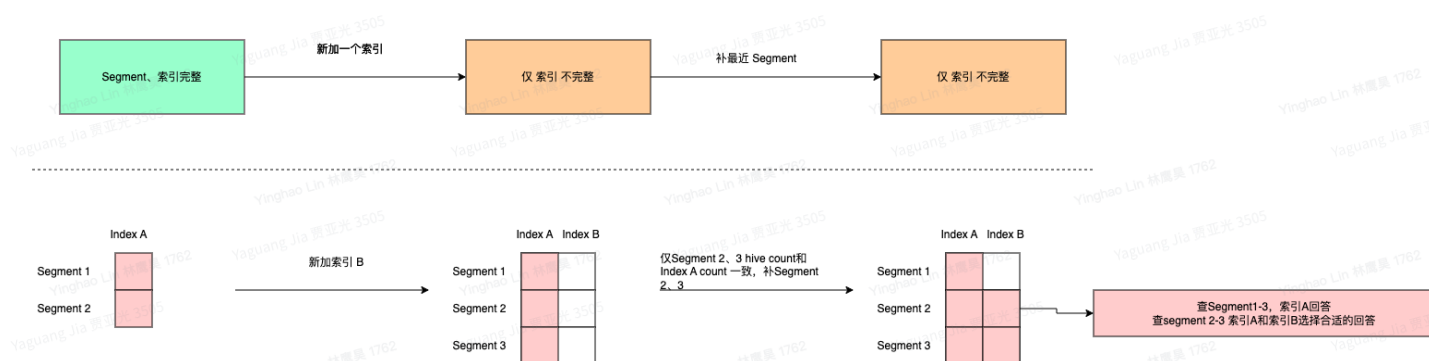
# KYLIN-5527 Kylin job engine adds the ability to check entries with data source(e.g. hive)

## Background

If the user periodically deletes the old data in the hive table, and some newly added indexes need to be complemented on the historical segment, then it is necessary to do

- The data should be accurate. In the case that the hive table data history exists and is now empty, the new index of the empty data cannot be filled out, resulting in inconsistent new and old index data.

Solution: Build a heterogeneous segment



KE has supported the construction and query of heterogeneous segments before, but the construction needs to manually identify empty segments and build them one by one, without automatic judgment for empty segments

In addition, when the historical data entry of the hive table is changed rather than deleted, when the data needs to be retrieved from the hive to build the index, there is currently no corresponding logic to ensure the consistency of the old and new index data in the model in this scenario

In summary, it is necessary to increase the data consistency check in the index complement process, solve the above problems of China Merchants Bank, and further enhance the identification and processing capabilities of KE products in the face of data inconsistency scenarios

# Dev Design

There are two main changes: index complement build adds Data Count Check and build task supports segment level partial success

## 1. Index Complement Build Add Data Count Check

If any of the above checks fails, the complement construction task of the segment is skipped directly.

- The count check logic is controlled by a model-level switch and is off by default

```
kylin.build.data-count-check-enabled
```

- Count check only for tasks with INDEX\_BUILD task type
  - The display type of the task interface of this type is: Build Index
  - 2 types of data ranges: full load, and segment time range

▶ 构建索引	<a href="#">model1</a>	全量加载
▶ 构建索引	<a href="#">test1</a>	2022-10-02 00:00:00...

- When the index complement needs to be carried out from hive
  - Execute check1 first
  - Execute check2 again (because the level table data will be generated first when complementing, so the actual check is the consistency of the existing index and the level table count value)
- When the index complement can be derived from the parent index
  - Execute check1 only
- Count check the position and do it in the generated flat table.

- 等待资源

持续时间: 1.6m

- 构建或刷新快照

持续时间: 0.04m

- 物化事实表视图

持续时间: < 0.01m

- 生成全局字典

持续时间: < 0.01m

- 生成平表

持续时间: < 0.01m

- 获取平表统计信息

持续时间: 0.02m

- 分层构建索引 3/3

持续时间: 0.05m

- 更新平表统计信息

持续时间: 0.65m

count check

- How to judge whether this index complement involves a flat table?

AdaptiveSpanningTree#fromFlatTable

```
public boolean fromFlatTable() {  
    return level0thNodes.stream().anyMatch(TreeNode::parentIsNull);  
}
```

- Count check the specific implementation?
  - There is a ready-made method for count checking of indexes

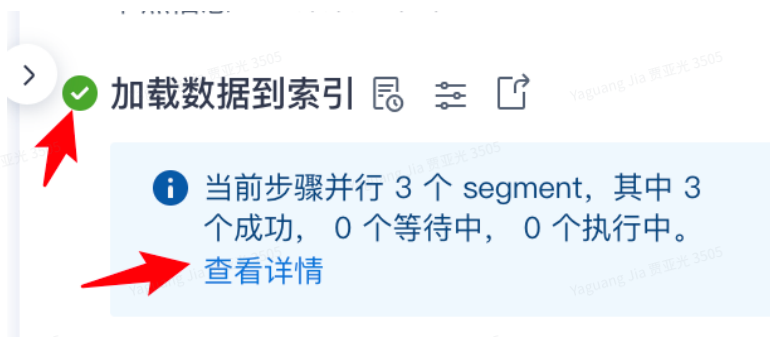
```
64 object SanityChecker extends LogEx {  
65  
66     val SKIP_FLAG: Long = -1L  
67  
68     def getCount(df: DataFrame, layout: LayoutEntity): Long = {  
69         if (IndexEntity.isTableIndex(layout.getId)) {  
70             df.count()  
71         } else {  
72             layout.getOrderedMeasures.values().asScala.find(m => m.getFunction.isCountConstant && !m.isTomb) match {  
73                 case Some(countMeasure) =>  
74                     val countMeasureCol = countMeasure.getId.toString  
75                     val row = df.select(countMeasureCol).agg(sum(col(countMeasureCol))).collect().head  
76                     if (row.isNullAt(0)) 0L else row.getLong(0)  
77                 case _ =>  
78                     SKIP_FLAG  
79             }  
80         }  
81     }  
82 }
```

- Flat table count check: direct count can be
- The count calculation needs to be called through slowStartExec to prevent congestion caused by uncontrolled submission of spark tasks

- Added AbnormalType in NDataLayout to represent the current layout exception

```
@JsonAutoDetect(fieldVisibility = JsonAutoDetect.Visibility.NONE,  
public class NDataLayout implements Serializable {  
  
    public enum AbnormalType {  
        DATA_INCONSISTENT  
    }  
}
```

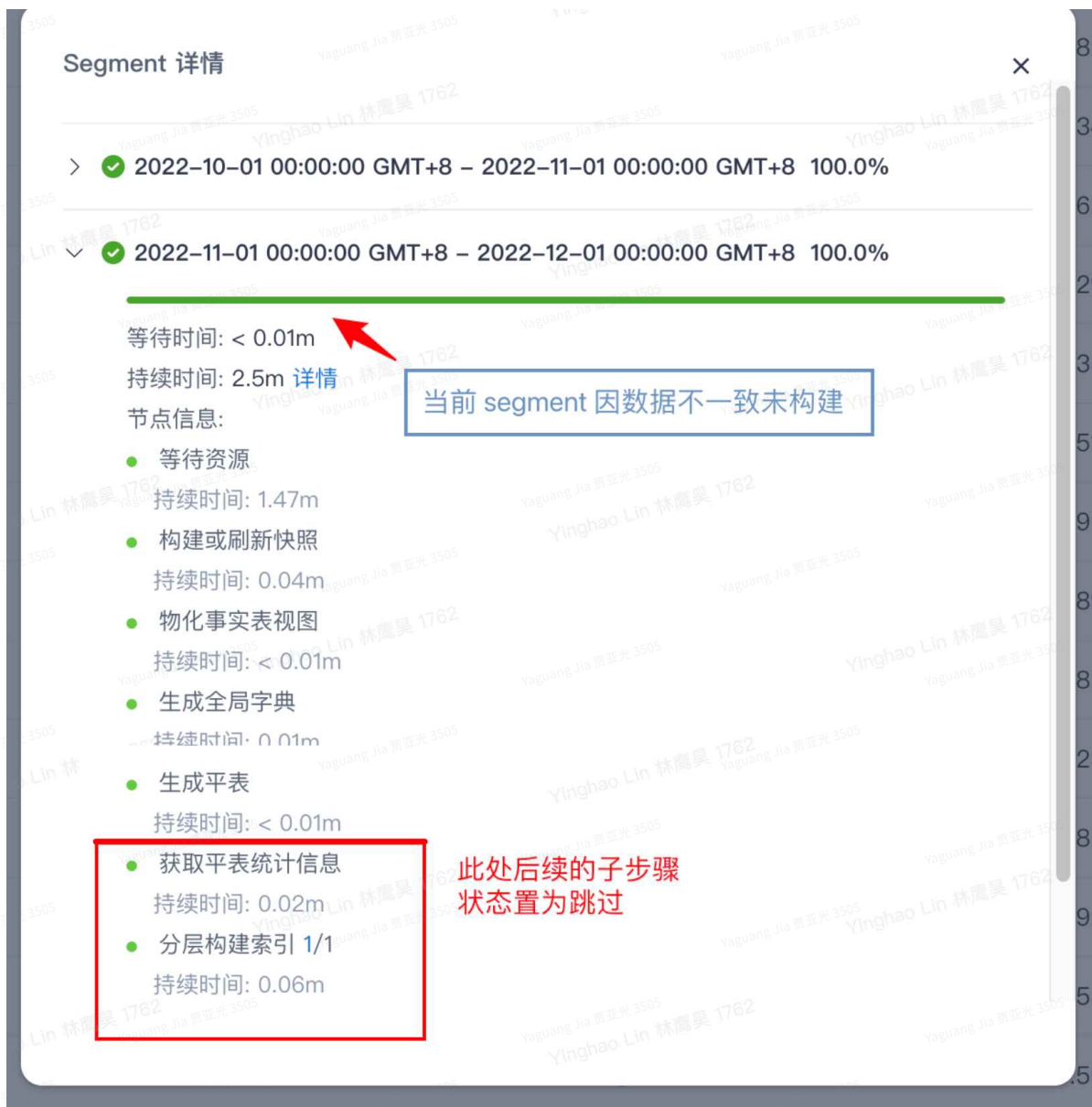
- But all segments must succeed for the mission to be successful
- Changed to
  - New sub-step status: WARNING
  - Allow segment level sections to succeed under count check mechanism
    - Single segment task, sub-step WARNING, the overall task is still FINISHED
    - Multiple segments of the same task are constructed in parallel, there is a segment task of WARNING, and the overall task is still FINISHED
  - Failure for other reasons (ERROR) will still cause the overall task to fail, which is consistent with the original product behavior
  - Successfully built segments, the index can be used
  - The prompt information is changed as follows:
    - "The current step has 3 segments in parallel, of which 2 are successful, 1 is not built due to data inconsistency, 0 are waiting, and 0 are executing"
    - If all segments are not built, the upper left corner green tick icon needs to be changed to a yellow triangle icon



- After clicking "View Details"



- After expanding an unbuilt segment



- If a task contains only one segment that skips building, what is the overall task status?

- The overall mission status remains FINISHED



- But the sub-step status is as follows:

✓ 加载数据到索引 1762

等待时间: < 0.01m

持续时间: 2.08m 详情

节点信息: 127.0.0.1:7070

● 等待资源

持续时间: 0.89m

● 构建或刷新快照

持续时间: 0.2m

● 物化事实表视图

持续时间: < 0.01m

● 生成全局字典

持续时间: 0.04m

● 生成平表

持续时间: < 0.01m

● 获取平表统计信息

持续时间: 0.01m

● 分层构建索引 1/1

持续时间: 0.07m

● 更新平表统计信息

持续时间: 0.85m

当前 segment 因数据不一致未构建

这3个步骤  
状态置为跳过

- Sub-step effect
  - 获取平表统计信息  
持续时间: –
  - 分层构建索引 0/1  
持续时间: –
  - 更新平表统计信息  
持续时间: –
- Other tasks contain a skipped segment, can they be submitted?
  - As long as the segment is in the construction process, other tasks cannot be submitted if they contain the segment
- A segment with only one exception index (which contains only one exception index), what is the status of the segment?



- Status as shown in the figure

<div> <div>基本消息</div> <div>数据特征</div> <div>Segment</div> <div>索引</div> <div>开发者</div> </div>	<div>Segment 列表</div> <div>刷新</div> <table> <tr> <th><input type="checkbox"/></th><th>开始时间</th><th>结束时间</th><th>索引数</th><th>状态</th><th>最</th></tr> <tr> <td><input type="checkbox"/></td><td>2022-10-02 00:00:00 GMT+8</td><td>2022-10-05 00:00:00 GMT+8</td><td>2/4</td><td>ONLINE</td><td>21</td></tr> <tr> <td><input type="checkbox"/></td><td>2022-10-01 00:00:00 GMT+8</td><td>2022-10-02 00:00:00 GMT+8</td><td>0/4</td><td>ONLINE</td><td>21</td></tr> </table>					<input type="checkbox"/>	开始时间	结束时间	索引数	状态	最	<input type="checkbox"/>	2022-10-02 00:00:00 GMT+8	2022-10-05 00:00:00 GMT+8	2/4	ONLINE	21	<input type="checkbox"/>	2022-10-01 00:00:00 GMT+8	2022-10-02 00:00:00 GMT+8	0/4	ONLINE	21
<input type="checkbox"/>	开始时间	结束时间	索引数	状态	最																		
<input type="checkbox"/>	2022-10-02 00:00:00 GMT+8	2022-10-05 00:00:00 GMT+8	2/4	ONLINE	21																		
<input type="checkbox"/>	2022-10-01 00:00:00 GMT+8	2022-10-02 00:00:00 GMT+8	0/4	ONLINE	21																		

- This status quo

- About segment coverage mechanism [KE-39001 CLONE - KE4构建支持segment覆盖 Acceptance](#)

- Segment coverage builds all indexes (including new indexes that have not been complemented before)
- The consistency of all index data in this segment can be guaranteed, so this change will not be affected

### 3. Metadata changes

- Current situation

```

"layout_instances": [
  {
    "layout_id": 20000000001,
    "build_job_id": "d76eab2c-8345-505f-cdbe-ebfe5e377c0b-fac43c52-d306-62",
    "rows": 11,
    "byte_size": 6346,
    "file_count": 1,
    "source_rows": 11,
    "source_byte_size": 0,
    "partition_num": 1,
    "partition_values": [],
    "is_ready": false,
    "create_time": 1670818040131,
    "multi_partition": []
  },
  {
    "layout_id": 10001,
    "build_job_id": "35262a3c-2388-6bf2-61b6-b02885868cd0-fac43c52-d306-62",
    "rows": 8,
    "byte_size": 2124,
    "file_count": 1,
    "source_rows": 11,
    "source_byte_size": 0,
    "partition_num": 1,
    "partition_values": [],
    "is_ready": false,
    "create_time": 1670818946166,
    "multi_partition": []
  },
  {
    "layout_id": 20000010001,
    "build_job_id": "f277d7e1-7e44-213c-5ae5-27d7d38dcd74-fac43c52-d306-62",
    "rows": 11
  }
]

```

- New design, add abnormal\_type field to indicate layout exception



```

{
  "layout_id": 20001,
  "build_job_id": "4c20f2c7-7b2f-0504-6db7-e5d921fd993f",
  "rows": 0,
  "byte_size": 0,
  "file_count": 0,
  "source_rows": 0,
  "source_byte_size": 0,
  "partition_num": 0,
  "partition_values": [],
  "is_ready": false,
  "create_time": 1675757603265,
  "multi_partition": [],
  "abnormal_type": "DATA_INCONSISTENT"
},
{

```

Since the current layout\_instances are stored in the effective layouts, and the new design needs to add layouts state, but also to join the non-effective state of the layouts, so the following code uses the layout\_instances place according to the different needs of the context semantics:

- Semantics 1: Only need to get valid layouts
- Semantics 2: Need to get all layouts

Method **getLayouts()** of `org.apache.kylin.metadata.cube.model.NDataSegDetails` 21+ usages

Project production files

File	Line	Code Snippet
NDataLayout.java	263	for (NDataLayout cached : segDetails.getLayouts()) {
NDataSegDetails.java	117	for (NDataLayout cuboid : getLayouts()) {
NDataSegDetails.java	128	for (NDataLayout cuboid : getLayouts()) {
NDataSegDetails.java	159	List<NDataLayout> currentSortedLayouts = getSortedLayouts(getLayouts());
NDataSegDetails.java	160	List<NDataLayout> anotherSortedLayouts = getSortedLayouts(another.getLayouts());
NDataSegment.java	347	List<NDataLayout> filteredCuboids = segDetails.getLayouts().stream()
NDataSegment.java	353	List<NDataLayout> cuboids = segDetails.getLayouts();
NDataflowManager.java	733	if (seg.getStatus() == SegmentStatusEnum.WARNING && segDetails != null && segDetails
NDataflowManager.java	833	val layouts = segment.getSegDetails().getLayouts();
SegmentPartition.java	145	.getLayouts().stream() //
NDataSegmentResponse.java	130	long segmentFileCount = segment.getSegDetails().getLayouts().stream()
IndexPlanService.java	730	if ((seg.getSegDetails().getLayouts().size() - lockedIndexCountInSeg) != allIndexCountWi
ModelService.java	1183	val segLayoutIds = segment.getSegDetails().getLayouts().stream().map(NDataLayout::ge
ModelService.java	2393	List<NDataLayout> layouts = new LinkedList<>(segDetails.getLayouts());
PartitionDictionaryBuilderHelper.java	52	for (NDataLayout cuboid : seg.getSegDetails().getLayouts()) {
AfterBuildResourceMerger.java	112	dfUpdate.setToAddOrUpdateLayouts(theSeg.getSegDetails().getLayouts().toArray(new N
AfterMergeOrRefreshResourceMerger.java	83	toUpdateCuboids.addAll(new ArrayList<>(mergedSegment.getSegDetails().getLayouts())
AfterMergeOrRefreshResourceMerger.java	158	toUpdateCuboids.addAll(new ArrayList<>(mergedSegment.getSegDetails().getLayouts())
DictionaryBuilderHelper.java	141	for (NDataLayout cuboid : seg.getSegDetails().getLayouts()) {
DFMergeJob.java	76	for (NDataLayout cuboid : seg.getSegDetails().getLayouts()) {
DataflowCleanerCLI.java	81	toBeRemoved.addAll(segment.getSegDetails().getLayouts().stream().map(NDataLayout::

...<15 usages are out of scope 'Project production files'>...

Press `⌘F7` again to search in 'Project Files'

Document	Lines of code	The purpose of using the getLayouts () method

NDataLayout.java	263	Used in the isCachedAndShared () method to determine the NDataLayout has been cached before updating the p
NDataSegDetails.java	117	getTotalRowCount method, count row counts
NDataSegDetails.java	128	getLayoutByld method, get layout according to id numb
NDataSegDetails.java	159	Determine whether the layout is consistent before Segm
NDataSegDetails.java	160	Determine whether the layout is consistent before Segm
NDataSegment.java	347	Filter the null NDataLayout and re-assign the content to i in the segment
NDataSegment.java	353	Put segment layouts into cached layoutsMap
NDataflowManager.java	733	After clearing all layouts in the warning status segment, t changes to ready
NDataflowManager.java	833	Delete multi-level partition information in layout
SegmentPartition.java	145	Calculate multi-level partitioned data capacity
NDataSegmentResponse.java	130	Determine whether there is a valid underlying detail inde
IndexPlanService.java	730	Calculate the number of segments that need to complet
ModelService.java	1183	Filter out segments with incomplete index construction.
ModelService.java	2393	In the delete index logic, filter and judge to delete the corresponding layout in each segment.
PartitionDictionaryBuilderHelper. java	52	Multilevel partition model index supplements global dict
AfterBuildResourceMerger.java	112	After the incremental build is completed, merge Metadat the layout information to count the layout size
AfterMergeOrRefreshResourceMer ger.java	83	After merging or refreshing, merge Metadata to get the m partition model segment layout information and statistic size
AfterMergeOrRefreshResourceMer ger.java	158	After merging or refreshing, merge Metadata to get the g model segment layout information and statistical layout
DictionaryBuilderHelper.java	141	Regular model index supplements global dictionary
DFMergeJob.java	76	Resource detection steps before segment merging to obt for subsequent reading of parquet files

MergeStage.scala	81	Segment merge, get layouts, submit to spark for merge c
PartitionMergeStage.scala	53	Multi-level partition model segment merge, get layouts, : spark for merge operation
DataflowCleanerCLI.java	81	Command line tool to clean up layouts that do not exist i plan in the segment

## 4. Rest API changes

- New, get the index list under segment, interface

Extend from existing interfaces: `GET /api/index_plans/index? project = & segment_id =`

Among them, `segment_id` is this new field

The original interface is not open in the manual, so it will not affect the use of existing customers

```
@ApiOperation(value = "getIndex", tags = { "AI" }, notes = "Update response: total_size")
@GetMapping(value = "/index")
public EnvelopeResponse<FusionRuleDataResult<List<IndexResponse>>> getIndex(
    @RequestParam(value = "project") String project, @RequestParam(value = "model") String modelId, //
    @RequestParam(value = "sort_by", required = false, defaultValue = "") String order,
    @RequestParam(value = "reverse", required = false, defaultValue = "false") Boolean desc,
    @RequestParam(value = "sources", required = false, defaultValue = "") List<IndexEntity.Source> sources,
    @RequestParam(value = "key", required = false, defaultValue = "") String key,
    @RequestParam(value = "status", required = false, defaultValue = "") List<IndexEntity.Status> status,
    @RequestParam(value = "ids", required = false, defaultValue = "") List<Long> ids,
    @RequestParam(value = "page_offset", required = false, defaultValue = "0") Integer offset,
    @RequestParam(value = "page_size", required = false, defaultValue = "10") Integer limit,
    @RequestParam(value = "range", required = false, defaultValue = "") List<IndexEntity.Range> range) {
    checkProjectName(project);
```

## 5. Impact on other scenes

- Full build task: data count check is required when indexing complement
- Split into multiple build tasks: no effect, only indexes passed by count check will be built, regardless of how segment is built

## 构建索引

该索引未构建至以下数据范围。为了提高查询效率，建议您将该索引构建至：

<input checked="" type="checkbox"/>	开始时间 ▾	结束时间 ▾
<input checked="" type="checkbox"/>	2022-10-01 00:00:00 GMT+8	2022-10-02 00:00:00 GMT+8
<input checked="" type="checkbox"/>	2022-10-02 00:00:00 GMT+8	2022-10-05 00:00:00 GMT+8

共 2 条

☒ 拆分多个任务并发构建 ①



- Construction of new model: no effect, no existing index is not affected by count check
- Segment merge: No effect, index inconsistencies cannot be merged.
- Segment refresh: does not affect, refresh that is, all index data is built from hive, and count check is meaningless

## 6. Special Case Description

- In the China Merchants Bank 3L 4 scenario, the base cuboid is used as both the basic aggregation and the basic detail.

Solution: Add switch, allow, detail, and aggregate, the count value is inconsistent between indexes

Switches can be global, project, model level:

- Off by default, meaning that all index counts must be consistent
- If turned on, count inconsistencies between detail and aggregation are allowed, but count between detail and detail, aggregation and aggregation still need to be consistent.
- `kylin.build.allow-non-strict-count-check`

## 7. Legacy problem

- Whether the switch is turned on by default or not requires performance verification
  - Local count operation on small data volume index, one is about 300ms
  - The current specification test only covers the full build scenario 国基准四, tpch100, 22sql自动建模+大 view requirements