

# KYLIN-5437 After the build job is terminated, the total time spent on the job is still calculated

## Root Cause:

```
"version": "4.0.0.0",
"name": "Waiting for yarn resources",
"tasks": null,
"type": "io.kyligence.kap.engine.spark.job.st

"handler_type": null,
"params": {
  "parentId": "7a31ab86-a4b7-b13c-6b25-24021

[
  },
  "segments": [
    ],
    "job_type": "STAGE",
    "data_range_start": 0,
    "data_range_end": 0,
    "target_model": null,
    "target_segments": [
      ],
      "output": {
        "content": null,
        "log_path": null,
        "status": "RUNNING",
        "info": {
```

```
id:7a31ab86)-4662] execution.NExecutableManager : UnicomWork 65566615 6676 5638 6288 4653131167 started on project project_66666666
id:7a31ab86)-4662] execution.NExecutableManager : [UNEXPECTED THINGS HAPPENED] wrong job state transfer! There is no valid state transfer from: DISCARDED to: RUNNING
id:7a31ab86)-4662] execution.NExecutableManager : Job id: 7a31ab86-a4b7-b13c-6b25-2402b8587748-8afc74d2-b0ca-429f-5caf-a4f8bda76599_01_00 from DISCARDED to RUNNING
```

From the task Metadata and log can be found, Stage 01 state illegal transition from **DISCARDED** to **RUNNING**, although the overall state of the Job at this time has been set to **DISCARDED**, but in the calculation of the total time will traverse the stage under the Job, if the stage state is Running, it will keep calculating the total time, that is, the following code, if the state of a stage is running, then the total time will continue to increase.

```

public static long getDuration(Output output) {
    if (output.getDuration() != 0) {
        var duration : long = output.getDuration();
        if (ExecutableState.RUNNING == output.getState()) {
            duration = duration + System.currentTimeMillis() - output.getLastRunningStartTime();
        }
        return duration;
    }
    if (output.getStartTime() == 0) {
        return 0;
    }
    return output.getEndTime() == 0 ? System.currentTimeMillis() - output.getStartTime()
        : output.getEndTime() - output.getStartTime();
}

```

> 任务 ID: 7a31ab86-a4b7-b13c-6b25-2402b858774  
8-8afc74d2-b0ca-429f-5caf-a4f8bda76599

对象: KYLIN JY AS JYKC\_STOCK\_PACK\_TRACE

状态: DISCARDED

等待时间: 0.79m

持续时间: 377h 41.53m

任务详情

⊖ 检测资源 ⓘ ≡

等待时间: 0.78m

持续时间: 0.94m 详情

节点信息: 192.168.45.123

⊖ 加载数据到索引 ⓘ ≡

等待时间: < 0.01m

持续时间: 377h 40.58m 详情

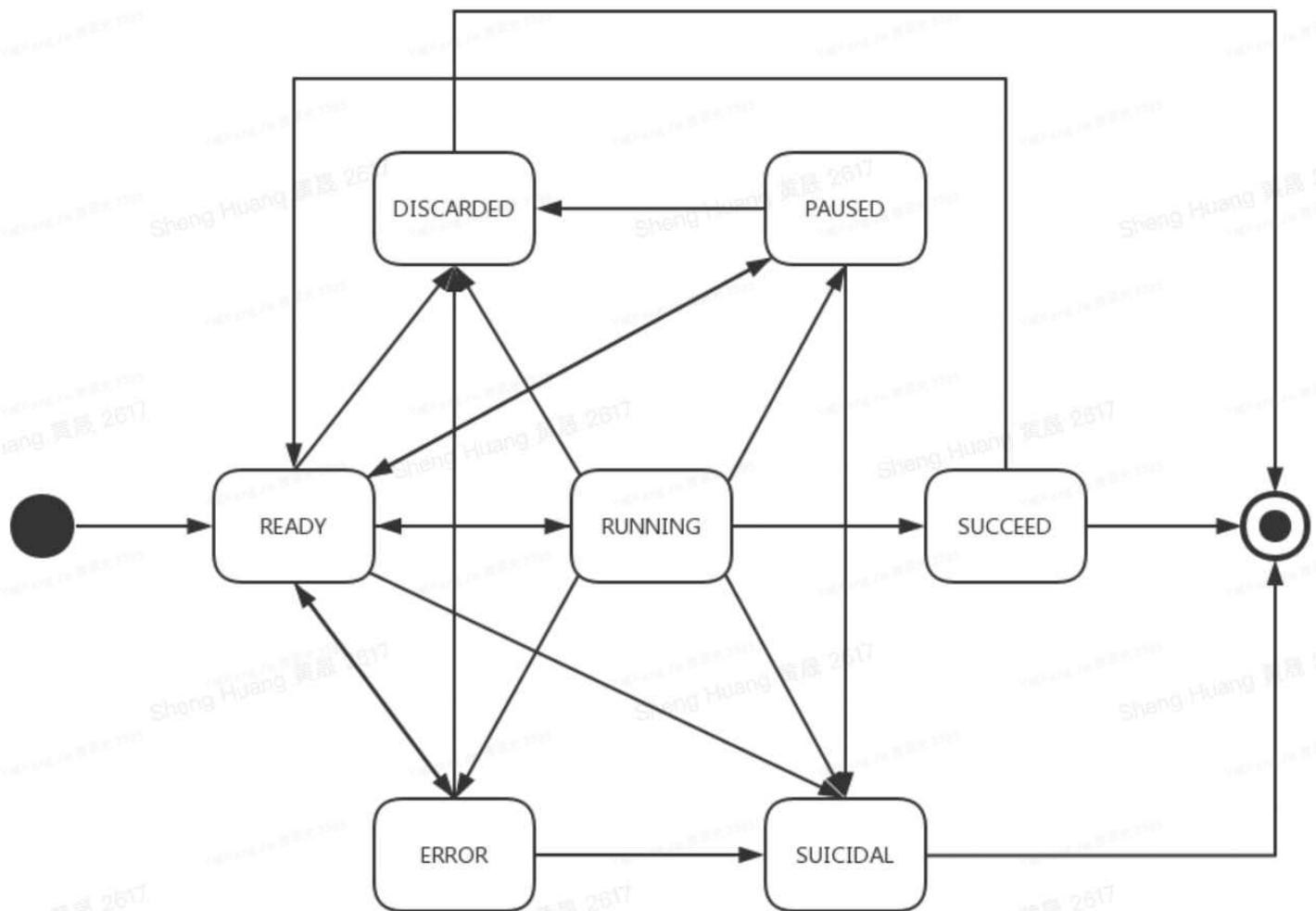
节点信息: 192.168.45.123

— 等待资源

持续时间: 377h 40.58m

— 构建或刷新数据

Fix Design:



Divided into two steps:

1. Finite-State Machine can be found, `DISCARDED` can not be converted to any other state, so you can directly from the source to prevent the occurrence of such illegal state transition in Stage, by determining whether the old state is `DISCARDED` , if so, directly return without any state transition. (consistent with the existing product behavior)
2. If the parent task state of a stage is `DISCARDED` , no matter what state the stage is in, it will be processed according to `DISCARDED` , that is, the total time will not be accumulated.

# Root Cause:

```
"version": "4.0.0.0",
"name": "Waiting for yarn resources",
"tasks": null,
"type": "io.kyligence.kap.engine.spark.job.sta

"handler_type": null,
"params": {
  "parentId": "7a31ab86-a4b7-b13c-6b25-24021

[",
},
"segments": [
],
"job_type": "STAGE",
"data_range_start": 0,
"data_range_end": 0,
"target_model": null,
"target_segments": [
],
"output": {
  "content": null,
  "log_path": null,
  "status": "RUNNING",
  "info": {
```

```
id:7a31ab86-4662] transactionMonitor : monitor 6556825-608-3c5d-62ad-7d371377b7 started on project project_sddadp1
id:7a31ab86-4662] execution.NExecutableManager : [UNEXPECTED THINGS HAPPENED] wrong job state transfer! There is no valid state transfer from: DISCARDED to: RUNNIN
id:7a31ab86-4662] execution.NExecutableManager : Job id: 7a31ab86-a4b7-b13c-6b25-2402b8587748-8afc74d2-b0ca-429f-5caf-a4f8bda76599_01_00 from DISCARDED to RUNNING
```

从任务元数据和日志可以发现，Stage 01 的状态非法的从 DISCARDED 转换为了 RUNNING，虽然此时该Job的整体状态已经置为 DISCARDED，但是在计算总耗时的时候会遍历Job下的stage，如果Stage的状态为Running的话就会不停的计算总耗时，即下面这段代码，如果一个stage的状态为runing，那么总耗时则会不断的增加。

```
public static long getDuration(Output output) {
    if (output.getDuration() != 0) {
        var duration : long = output.getDuration();
        if (ExecutableState.RUNNING == output.getState()) {
            duration = duration + System.currentTimeMillis() - output.getLastRunningStartTime();
        }
        return duration;
    }
    if (output.getStartTime() == 0) {
        return 0;
    }
    return output.getEndTime() == 0 ? System.currentTimeMillis() - output.getStartTime()
        : output.getEndTime() - output.getStartTime();
}
```

> 任务 ID: 7a31ab86-a4b7-b13c-6b25-2402b858774  
8-8afc74d2-b0ca-429f-5caf-a4f8bda76599

对象: KYLIN JY AS JYKC\_STOCK\_PACK\_TRACE

状态: DISCARDED

等待时间: 0.79m

持续时间: 377h 41.53m

### 任务详情

#### 检测资源

等待时间: 0.78m

持续时间: 0.94m [详情](#)

节点信息: 192.168.45.123

#### 加载数据到索引

等待时间: < 0.01m

持续时间: 377h 40.58m [详情](#)

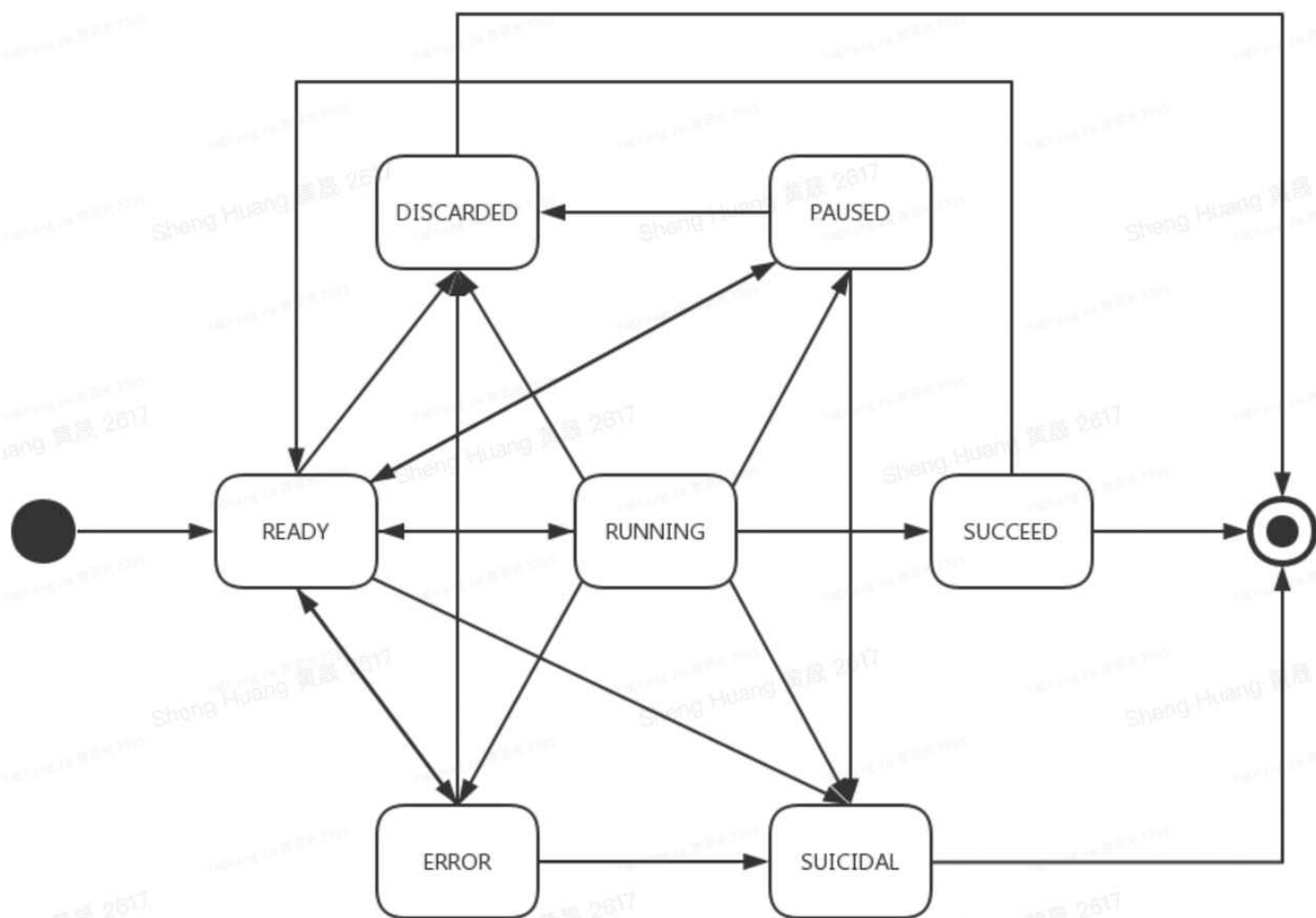
节点信息: 192.168.45.123

#### 等待资源

持续时间: 377h 40.58m

构建或刷新索引

Fix Design:



分为两步：

1. 通过状态机可以发现，DISCARDED 无法转换成其他任何状态，所以可以直接从源头上杜绝在 Stage 中发生此类非法状态的转换，通过判断旧的状态是否为 DISCARDED，如果是那么直接返回不进行任何状态转换。（与现有产品行为一致）
2. 在总耗时计算逻辑上进行调整，如果一个stage的父任务状态为 DISCARDED，无论该stage为任何状态都一律按照 DISCARDED 处理，即不再累计计算总耗时。