

Spark 任务失败会影响accumulator的准确性吗？

📅 发布于 2021-03-09 | 📁 分类于 Spark | ⌚ 3分钟 | 📖 469字数

通常在使用Spark进行数据处理时，为了高性能的统计某些全局指标我们会使用SparkContext.xxxAccumulator()来创建累加器，然后在具体的算子中进行累加，从而省略了执行特定的统计逻辑。

但是，这里需要考虑一个问题，如果某一个task失败后进行重试会不会因为重复统计导致统计指标不准确呢？

我们需要理解Spark对accumulator的实现后才能确定，下面就以SparkContext.longAccumulator()为例来看看事情的真相。

1.创建LongAccumulator

```
/**
 * Create and register a long accumulator, which starts with 0 and accumulates inputs by `add`.
 */
def longAccumulator: LongAccumulator = {
  val acc = new LongAccumulator
  register(acc)
  acc
}
```

[复制代码](#)

2.LongAccumulator的注册

```
private[spark] def register(
  sc: SparkContext,
  name: Option[String] = None,
  countFailedValues: Boolean = false): Unit = {
  if (this.metadata != null) {
    throw new IllegalStateException("Cannot register an Accumulator twice.")
  }
  this.metadata = AccumulatorMetadata(AccumulatorContext.newId(), name, countFailedValues)
  AccumulatorContext.register(this)
  sc.cleaner.foreach(_.registerAccumulatorForCleanup(this))
}
```

可以看到Accumulator注册参数中有一个countFailedValues参数默认为false，这里回答我们提出的问题：Spark的accumulator已经考虑到task失败的情况，且默认情况下失败的task不会计入accumulator，可以放心使用。

3.countfailedValues是如何生效的？

Executor在任务执行结束后会调用 `task.collectAccumulatorUpdates()` 对accumulator进行更新：

```
private[spark] class Executor(
  executorId: String,
  executorHostname: String,
  env: SparkEnv,
  userClassPath: Seq[URL] = Nil,
  isLocal: Boolean = false,
  uncaughtExceptionHandler: UncaughtExceptionHandler = new SparkUncaughtExceptionHandler)
extends Logging {
  .....

  class TaskRunner(
    execBackend: ExecutorBackend,
    private val taskDescription: TaskDescription)
```

```

extends Runnable {
    .....
    override def run(): Unit = {
        .....
        val accumUpdates = task.collectAccumulatorUpdates() //对accumulator进行更新
        .....
    }
}
}
}

```

而更新逻辑中会剔除失败任务上报的accumulator

```

def collectAccumulatorUpdates(taskFailed: Boolean = false): Seq[AccumulatorV2[_]] = {
    if (context != null) {
        // Note: internal accumulators representing task metrics always count failed values
        context.taskMetrics.nonZeroInternalAccums() ++
        // zero value external accumulators may still be useful, e.g. SQLMetrics, we should not
        // filter them out.
        context.taskMetrics.externalAccums.filter(a => !taskFailed || a.countFailedValues)
    } else {
        Seq.empty
    }
}
}

```

复制代码

最后Executor将accumulator数据作为task结果的一部分，序列化后写入到BlockManager中，Driver端读取到的就是不包含失败任务的accumulator数据了。

本文作者： Oleg

本文链接： <https://olegpt.top/post/spark-accumulator-zhen-de-zhun-que-ma/>

版权声明： 本博客所有文章除特别声明外，均采用© BY-NC-SA 许可协议。转载请注明出处！