

一. 概述

storm metric类似于hadoop的counter，用于收集应用程序中的特定指标，输出到外部。在storm中是存储到各个机器logs目录下的metric.log文件中。有时我们想保存一些计算的中间变量，当达到一定状态时，统一在一个位置输出，或者统计整个应用的一些指标，metric是个很好的选择。

二. 使用

1. 在bolt的prepare注册metric

```
transient CountMetric _countMetric;
transient MultiCountMetric _wordCountMetric;
transient ReducedMetric _wordLengthMeanMetric;

void initMetrics(TopologyContext context) {
    _countMetric = new CountMetric();
    _wordCountMetric = new MultiCountMetric();
    _wordLengthMeanMetric = new ReducedMetric(new MeanReducer());
    context.registerMetric("execute_count", _countMetric, 5);
    context.registerMetric("word_count", _wordCountMetric, 60);
    context.registerMetric("word_length", _wordLengthMeanMetric, 60);
}
```

- 1). metric都定义为了transient。因为这些Metric实现都没有实现Serializable，而在storm的spout/bolt中，所有非transient的变量都必须Serializable
- 2). 三个参数为metric名称，metric对象，以及时间间隔。时间间隔表示多久一次metric将数据发送给metric consumer。

2. 在bolt的execute方法中更新metric

```
void updateMetrics(String word) {
    _countMetric.incr();
    _wordCountMetric.scope(word).incr();
    _wordLengthMeanMetric.update(word.length());
}
```

3. 在topo中指定将metric consumer，这里使用了storm自带的consumer将其输出到日志文件中，也可以自定义consumer。

```
conf.registerMetricsConsumer(LoggingMetricsConsumer.class, 2);
```

三. 详细说明

1. metric和metric consumer

storm中关于metric的API主要有2部分：metric和metric consumer。前者用于生成一些统计值，后者将其处理后输出。

2. metric

- 1). metric在用于在spout/bolt中收集统计值。
- 2). metric通过TopologyContext.registerMetric(...)来注册到相应的spout或者bolt中，一般是在prepare/open方法中注册metric，然后在execute方法中更新metric值。
- 3). Metric必须实现backtype.storm.metric.api.IMetric接口。

3. 自带的Metric实现

storm提供了几个常用的metric实现：

- 1). AssignableMetric – set the metric to the explicit value you supply. Useful if it's an external value or in the case that you are already calculating the summary statistic yourself. Note: Useful for statsd Gauges.
- 2). CombinedMetric – generic interface for metrics that can be updated associatively.
- 3). CountMetric – a running total of the supplied values. Call incr() to increment by one, incrBy(n) to add/subtract the given number. Note: Useful for statsd counters.
- 4). MultiCountMetric– a hashmap of count metrics. Note: Useful for many Counters where you may not know the name of the metric a priori or where creating many Counter's manually is burdensome.
- 5). MeanReducer – an implementation of ReducedMetric that tracks a running average of values given to its reduce() method. (It accepts Double, Integer or Long values, and maintains the internal average as a Double.) Despite his reputation, the MeanReducer is actually a pretty nice guy in person.

4. Metric Consumer

- 1). MetricsConsumer用于收集拓扑中注册的所有Metric，并进行处理、输出等。处理的对象为DataPoint类的对象，同时包括了一些额外信息，如worker主机，worker端口，组件ID，taskID，时间戳，更新周期等。
- 2). MetricsConsumer使用backtype.storm.Config.registerMetricsConsumer(...)在创建topo时注册，或者在storm的配置文件中指定topology.metrics.consumer.register。
- 3). MetricsConsumer必须实现backtype.storm.metric.api.IMetricsConsumer接口。