

## Flink监控指标名特殊字符解决

和其他大数据系统类似，Flink 内置也提供 metric system 供我们监控 Flink 程序的运行情况，包括了JobManager、TaskManager、Job、Task以及Operator等组件的运行情况，大大方便我们调试监控我们的程序。

系统提供的一些监控指标名字有下面几个：

- metrics.scope.jm
  - 默认值: <host>.jobmanager
  - job manager范围内的所有metrics将会使用这个路径
- metrics.scope.jm.job
  - 默认值: <host>.jobmanager.<job\_name>
  - job manager 和 job范围内的所有metrics将会使用这个路径
- metrics.scope.tm
  - 默认值: <host>.taskmanager.<tm\_id>
  - task manager范围内的所有metrics将会使用这个路径
- metrics.scope.tm.job
  - 默认值: <host>.taskmanager.<tm\_id>.<job\_name>
  - task manager 和 job范围内的所有metrics将会使用这个路径
- metrics.scope.task
  - 默认值:  
<host>.taskmanager.<tm\_id>.<job\_name>.<task\_name>.<subtask\_index>
  - task 范围内的所有metrics将会使用这个路径
- metrics.scope.operator
  - 默认值:  
<host>.taskmanager.<tm\_id>.<job\_name>.<operator\_name>.<subtask\_index>
  - operator范围内的所有metrics将会使用这个路径

上面的<host>，<operator\_name>以及<subtask\_index>在程序运行的时候会替换成相应的值，比如<host>会替换成 iteblog.com，<operator\_name>会替换成FlatMapper等。



如果想及时了解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog\_hadoop

## 问题

现在假设我们有如下的程序

```
env.addSource(consumer)
    .flatMap(new IteblogFlatMapper(BLACK_TABLE, HASH_TABLE)).name("My Map")
    .addSink(new ElasticsearchSink[IteblogEntry](config, esServers,
        new IteblogElasticsearchSinkFunction, new IteblogActionRequestFailureHandler)).name("
es")
```

我们把这个程序的metric信息发送到Graphite监控系统中：

```
metrics.reporters: grph
metrics.reporter.grph.class: org.apache.flink.metrics.graphite.GraphiteReporter
metrics.reporter.grph.host: www.iteblog.com
metrics.reporter.grph.port: 2003
metrics.reporter.grph.protocol: TCP
```

当我们运行这个程序，相关的变量（<host>，<operator\_name>以及<subtask\_index>等）会替换成下面的值：

```
<job_id> -> 9e81f84c50820c304b8af2b16fa8140b
<task_id> -> cbc357ccb763df2852fee8c4fc7d55f2
<task_attempt_id> -> 690a6cebddd1bfa9edacfed50aa1d4807
<host> -> iteblog.com
<operator_name> -> Sink: es
<task_name> -> Source: Custom Source -> My Map -> Sink: es
<task_attempt_num> -> 0
<job_name> -> MyFlinkJobs
<tm_id> -> 34d31d54c0031328a6ec8910e571a7f8
<subtask_index> -> 0
```

所以metrics.scope.task（默认值：<host>.taskmanager.<tm\_id>.<job\_name>.<task\_name>.<subtask\_index>）范围指标的名字将会变成下面的路径：

flinkjobs.iteblog.com.taskmanager.34d31d54c0031328a6ec8910e571a7f8.MyFlinkJobs.Source: Custom Source -> TestFlat -> Map -> Sink: es.0

注意看<task\_name>变量的值为Source: Custom Source -> My Map -> Sink: es，而在Graphite系统中，其不支持空格、>以及:等特殊字符，因为指标名最终会以文件名的形式存储到文件系统中。metrics.scope.task范围内的所有指标将无法在Graphite中显示。

解决这个问题主要两两种方法：

- 修改metrics.scope.task属性的默认值；
- 修改GraphiteReporter类的相关代码。

## 修改metrics.scope.task属性的默认值

这种方法最简单了，针对本文的情况，我们可以不将<task\_name>变量写到指标名中，使得指标名中无特殊字符，比如我们做出如下修改（在 \$FLINK\_HOME/conf/flink-conf.yaml 文件里面修改）：

```
metrics.scope.task: <host>.taskmanager.<tm_id>.<job_name>.custom_task_name.<subtask_index>
```

这样运行上面程序的时候，metrics.scope.task属性最终的值如下：

```
flinkjobs.iteblog.com.taskmanager.34d31d54c0031328a6ec8910e571a7f8.MyFlinkJobs.custom_task_name.0
```

这样整个指标名中都无特殊字符，我们就可以在Graphite系统中看到相应的指标。

## 修改GraphiteReporter类的相关代码

上面那种方法我们将 <task\_name> 变量的值写死了，有时候这并不是我们想要的，所有这时候我们可以修改 GraphiteReporter类。GraphiteReporter类继承自 org.apache.flink.dropwizard.ScheduledDropwizardReporter，ScheduledDropwizardReporter类中有个 filterCharacters 函数，我们可以在这个函数里将特殊字符全部替换掉，下面是一种实现：

```
@Override
public String filterCharacters(String str) {
    char[] chars = null;
    final int strLen = str.length();
    int pos = 0;

    for (int i = 0; i < strLen; i++) {
        final char c = str.charAt(i);
        switch (c) {
            case '>':
            case '<':
            case '"':
                // remove character by not moving cursor
                if (chars == null) {
                    chars = str.toCharArray();
                }
                break;

            case '|':
                if (chars == null) {
                    chars = str.toCharArray();
                }
                chars[pos++] = '_';
                break;

            case ':':
            case '=':
            case ';':
            case ':':
            case '?':
            case 'W':
            case '*':
                if (chars == null) {
                    chars = str.toCharArray();
                }
                chars[pos++] = '-';
                break;

            default:
                if (chars != null) {
                    chars[pos] = c;
                }
                pos++;
            }
        }
    }
}
```

```
return chars == null ? str : new String(chars, 0, pos);  
}
```

我们把

- 指标名中的<, >以及" 特殊字符直接去掉了
- 指标名中的,, = , ; , : , ? , \w以及\* 特殊字符替换成-
- 指标名中的空格符替换成\_

然后编译相关的模块，并替换flink-metrics-graphite-1.3.1.jar，我们再运行上面的例子，这时候metrics.scope.task属性最终的值如下：

```
flinkjobs.iteblog.com.taskmanager.34d31d54c0031328a6ec8910e571a7f8.MyFlinkJobs.Source:  
Custom_Source_-_TestFlat_-_Map_-_Sink: es.0
```

这时候我们也可以在Graphite监控系统中看到相关的数据了。

**本博客文章除特别声明，全部都是原创！**  
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。  
本文链接: [【】（）](#)