Spark Metrics配置详解

和Hadoop类似，在Spark中也存在很多的Metrics配置相关的参数，它是基于Coda Hale Metrics Library的可配置Metrics系统，我们可以通过配置文件进行配置，通过Spark的Metrics系统，我们可以把Spark Metrics的信息报告到各种各样的Sink，比如HTTP、JMX以及CSV文件。

Spark的Metrics系统目前支持以下的实例：

- master：Spark standalone模式的master进程；
- worker：Spark standalone模式的worker进程；
- executor：Spark executor；
- driver：Spark driver进程；
- applications：master进程里的一个组件，为各种应用作汇报。

在Spark的Metrics系统主要支持Sink和Source两种，其中，Sink指定metrics信息发送到哪里，每个instance可以设置一个或多个Sink（这点和Flume很类似）。Sink的源码位于org.apache.spark.metrics.sink包中；而Source也是指信息的来源，它主要分为两大类：

- Spark内部source，比如MasterSource、WorkerSource等，它们会接收Spark组件的内部状态；
- 通用source，如：JvmSource，它收集低级别的状态。

支持的Sink类别

ConsoleSink

ConsoleSink是记录Metrics信息到Console中。

名称	默认值	描述
class	org.apache.spark.metrics.sink.ConsoleSink	Sink类
period	10	轮询间隔
unit	seconds	轮询间隔的单位

CSVSink

定期的把Metrics信息导出到CSV文件中。

名称	默认值	描述
class	org.apache.spark.metrics.sink.CsvSink	Sink类
period	10	轮询间隔
unit	seconds	轮询间隔的单位
directory	/tmp	CSV文件存储的位置

JmxSink

可以通过JMX方式访问Mertics信息

名称	默认值	描述
class	org.apache.spark.metrics.sink.JmxSink	Sink类

MetricsServlet

名称	默认值	描述
class	org.apache.spark.metrics.sink.MetricsServlet	Sink类
path	VARIES*	Path prefix from the web server root
sample	false	Whether to show entire set of samples for histograms ('false' or 'true') |

这个在Spark中默认就开启了，我们可以在4040端口页面的URL后面加上/metrics/json查看

GraphiteSink

名称	默认值	描述
class	org.apache.spark.metrics.sink.GraphiteSink	Sink类
host	NONE	Graphite服务器主机名
port	NONE	Graphite服务器端口
period	10	轮询间隔

名称	默认值	描述
unit	seconds	轮询间隔的单位
prefix	EMPTY STRING	Prefix to prepend to metric name

GangliaSink

由于Licene的限制，默认没有放到默认的build里面，如果需要使用，需要自己编译（这个会在后面专门介绍）

名称	默认值	描述
class	org.apache.spark.metrics.sink.GangliaSink	Sink类
host	NONE	Ganglia服务器的主机名或multicast group
port	NONE	Ganglia服务器的端口
period	10	轮询间隔
unit	seconds	轮询间隔的单位
ttl	1	TTL of messages sent by Ganglia
mode	multicast	Ganglia网络模式('unicast' or 'multicast')

如何使用

在Spark安装包的$SPARK_HOME/conf路径下有个metrics.properties文件（如果不存在，请将metrics.properties.template重命名为metrics.properties即可），Spark启动的时候会自动加载它。

当然，如果想修改配置文件位置，我们可以使用-Dspark.metrics.conf=xxx进行修改。

实例

下面我将简单地介绍如何使用Spark
Metrics。我只想简单地开启ConsoleSink，我们可以如下配置：

```
# User: 过往记忆
# Date: 2015-05-05
# Time: 上午01:16
```

```
# bolg:
# 本文地址：/archives/1341
# 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
# 过往记忆博客微信公共帐号：iteblog_hadoop

*.sink.console.class=org.apache.spark.metrics.sink.ConsoleSink
*.sink.console.period=10
*.sink.console.unit=seconds
```

period是ConsoleSink的轮询周期，unit是ConsoleSink的轮询周期时间单位。上面是配置所有的实例，如果想单独配置可以如下：

```
master.sink.console.class=org.apache.spark.metrics.sink.ConsoleSink
master.sink.console.period=15
master.sink.console.unit=seconds
```

这个配置可以覆盖通用配置符（也就是上面的*号）

我们为master、worker、driver和executor开启jvm source，如下：

```
# User: 过往记忆
# Date: 2015-05-05
# Time: 上午01:16
# bolg:
# 本文地址：/archives/1341
# 过往记忆博客，专注于hadoop、hive、spark、shark、flume的技术博客，大量的干货
# 过往记忆博客微信公共帐号：iteblog_hadoop

master.source.jvm.class=org.apache.spark.metrics.source.JvmSource
worker.source.jvm.class=org.apache.spark.metrics.source.JvmSource
driver.source.jvm.class=org.apache.spark.metrics.source.JvmSource
executor.source.jvm.class=org.apache.spark.metrics.source.JvmSource
```

当然，我们还可以自定义Source，这个需要继承自 org.apache.spark.metrics.source.Source 类。关于如何自定义Source，我这里不介绍了，需要的同学可以去参照Spark源码，比如 JvmSource 类的实现。