




Hadoop集群监控: jmx信息获取

 Hadoop  2016-06-23 21:06:49  5071  0评论 [下载为PDF](#)

Hadoop集群的监控可以通过多种方式来实现(比如REST API、jmx、内置API等等)。虽然监控方式有多种,但是我们需要根据监控的指标选择不同的监控方式,比如如果你想监控作业的情况,那么你选择jmx是不能满足的;你想监控各节点的运行情况,REST API也是不能满足的。所以在选择不同当时监控时,我们需要详细了解需要我们的需求。本文只介绍Hadoop的jmx监控。

Hadoop的jmx提供了诸如Cluster、Queue、Jvm、FSQueue等Metrics信息,而且获取它也是非常地方便。比如想获取YARN相关的jmx,我们可以在浏览器输入

<https://www.iteblog.com:8088/jmx> 即可获取一堆Jmx信息;同理,如果想获取NameNode的jmx可以输入 <https://www.iteblog.com:50070/jmx>。

/jmx 这个URL是通过 org.apache.hadoop.jmx.JMXJsonServlet 类实现的。一般情况下所有使用HttpServer启动服务都可以访问 <http://.../jmx> 链接。这个Servlet仅仅提供了JMX metrics只读权限。为了方便,这里只介绍YARN的jmx。

默认情况下只输入 <https://www.iteblog.com:8088/jmx> 将返回所有关于YARN的Metrics信息,并且是以Json格式返回的。比如下面的数据:

```
{
  "beans" : [ {
    "name" : "Hadoop:service=ResourceManager,name=FSOpDurations",
    "modelerType" : "FSOpDurations",
    "tag.FSOpDurations" : "FSOpDurations",
    "tag.Context" : "fairscheduler-op-durations",
    "tag.Hostname" : "www.iteblog.com",
    "ContinuousSchedulingRunNumOps" : 0,
    "ContinuousSchedulingRunAvgTime" : 0.0,
    "NodeUpdateCallNumOps" : 156794111,
    "NodeUpdateCallAvgTime" : 0.012121212121212114,
    "UpdateThreadRunNumOps" : 7045403,
    "UpdateThreadRunAvgTime" : 0.0,
    "UpdateCallNumOps" : 7045403,
    "UpdateCallAvgTime" : 0.0,
    "PreemptCallNumOps" : 0,
    "PreemptCallAvgTime" : 0.0
  } ],
  .....//这里省略了很多信息
}
```

从上面的结果可以看出,如果仅仅输入了 <http://.../jmx> 链接,服务器将返回一大推的Jmx信息,这其中肯定有很多不是我们想要的。如果我们仅仅需要获取我们想要的信息,比如我们想获取某个队列的Jmx信息,怎么办呢?从源码中我们可以了解到其实 org.apache.hadoop.jmx.JMXJsonServlet 类支持三个参数: callback、qry、get。下面分别介绍三个参数的含义。

文章目录

- 1 callback
- 2 qry
- 3 get

callback

callback 参数是用于需要JSONP响应的请求。JSONP(JSON with Padding)是JSON的一种“使用模式”,可用于解决主流浏览器的跨域数据访问的问题。使用这个参数之后HTTP头的ContentType变成了 application/javascript; charset=utf8,并且返回的Json被 callback 参数的值和()包围了。比如我访问的URL为: <https://www.iteblog.com:8088/jmx?callback=iteblog>,返回的结果如下:

```
iteblog({
  "beans" : [ {
    "name" : "Hadoop:service=ResourceManager,name=FSOpDurations",
    "modelerType" : "FSOpDurations",
    "tag.FSOpDurations" : "FSOpDurations",
    "tag.Context" : "fairscheduler-op-durations",
    "tag.Hostname" : "www.iteblog.com",
    "ContinuousSchedulingRunNumOps" : 0,
    "ContinuousSchedulingRunAvgTime" : 0.0,
    "NodeUpdateCallNumOps" : 156794111,
    "NodeUpdateCallAvgTime" : 0.012121212121212114,
    "UpdateThreadRunNumOps" : 7045403,
    "UpdateThreadRunAvgTime" : 0.0,
    "UpdateCallNumOps" : 7045403,
```

```

    "UpdateCallAvgTime" : 0.0,
    "PreemptCallNumOps" : 0,
    "PreemptCallAvgTime" : 0.0
  },
  ....//这里省略了很多信息
]
})

```

qry

上面也提到了，如果什么参数都不输入，那么服务器将返回一大堆的jmx信息，里面有很多都不是我们需要的。如果我们想获取某个队列的Jmx信息，怎么办呢？这就得借助 qry 参数了。这个参数接收jmx中name的值。比如我们想获取iteblog队列的jmx信息，可以这么请求：<https://www.iteblog.com:8088/jmx?qry=Hadoop:service=ResourceManager,name=QueueMetrics,q0=root,user=iteblog>，下面是其返回的信息：

```

{
  "beans" : [ {
    "name" : "Hadoop:service=ResourceManager,name=QueueMetrics,q0=root,user=iteblog",
    "modelerType" : "QueueMetrics,q0=root,user=iteblog",
    "tag.Queue" : "root",
    "tag.User" : "iteblog",
    "tag.Context" : "yarn",
    "tag.Hostname" : "www.iteblog.com",
    "running_0" : 0,
    "running_60" : 0,
    "running_300" : 0,
    "running_1440" : 0,
    "AppsSubmitted" : 355,
    "AppsRunning" : 0,
    "AppsPending" : 0,
    "AppsCompleted" : 353,
    "AppsKilled" : 2,
    "AppsFailed" : 0,
    "AllocatedMB" : 0,
    "AllocatedVCores" : 0,
    "AllocatedContainers" : 0,
    "AggregateContainersAllocated" : 461223,
    "AggregateContainersReleased" : 461223,
    "AvailableMB" : 0,
    "AvailableVCores" : 0,
    "PendingMB" : 0,
    "PendingVCores" : 0,
    "PendingContainers" : 0,
    "ReservedMB" : 0,
    "ReservedVCores" : 0,
    "ReservedContainers" : 0,
    "ActiveUsers" : 0,
    "ActiveApplications" : 3
  } ]
}

```

当然，我们也可以指定通配符，比如 https://www.iteblog.com:8088/jmx?qry=Hadoop:* 将会返回所有name为 Hadoop: 开头的jmx信息，如下：

```

{
  "beans" : [ {
    "name" : "Hadoop:service=ResourceManager,name=FSOpDurations",
    "modelerType" : "FSOpDurations",
    "tag.FSOpDurations" : "FSOpDurations",
    "tag.Context" : "fairscheduler-op-durations",
    "tag.Hostname" : "www.iteblog.com",
    "ContinuousSchedulingRunNumOps" : 0,
    "ContinuousSchedulingRunAvgTime" : 0.0,
    "NodeUpdateCallNumOps" : 157783485,
    "NodeUpdateCallAvgTime" : 0.030120481927710847,
    "UpdateThreadRunNumOps" : 7103943,
    "UpdateThreadRunAvgTime" : 0.09999999999999999,
    "UpdateCallNumOps" : 7103943,
    "UpdateCallAvgTime" : 0.09999999999999999,
    "PreemptCallNumOps" : 0,
    "PreemptCallAvgTime" : 0.0
  }, {
    "name" : "Hadoop:service=ResourceManager,name=UgiMetrics",
    "modelerType" : "UgiMetrics",
    "tag.Context" : "ugi",
    "tag.Hostname" : "www.iteblog.com",
    "LoginSuccessNumOps" : 0,

```

```
"LoginSuccessAvgTime" : 0.0,
"LoginFailureNumOps" : 0,
"LoginFailureAvgTime" : 0.0,
"GetGroupsNumOps" : 0,
"GetGroupsAvgTime" : 0.0
},
....
}}
```

如果查询的添加没有找到, 将会返回 {}。在Hadoop内部, 这个参数的实现是由MBeanServer类的 `public Set queryNames(ObjectName name, QueryExp query)` 方法实现的。

get

如果我们想获取jmx某个属性的值, 而不是一堆信息, 那么我们可以使用 `get` 参数。这个参数的值要求是 `MBeanName::AttributeName` 格式的。比如我们想获取某个队列的 `tag.Hostname` 属性的值, 我们可以这么请

求: `https://www.iteblog.com:8088/jmx?qry=Hadoop:service=ResourceManager,name=QueueMetrics,q0=root,user=iteblog::tag.Hostname`, 它的返回值为:

```
{
  "beans" : [ {
    "name" : "Hadoop:service=ResourceManager,name=QueueMetrics,q0=root,user=iteblog",
    "modelerType" : "QueueMetrics,q0=root,user=iteblog",
    "tag.Hostname" : "www.iteblog.com"
  } ]
}
```

如果`get`参数的值格式不对, 服务器将返回异常信息:

```
{
  "result" : "ERROR",
  "message" : "query format is not as expected."
}
```

我们可以编写重新解析这些jmx信息, 从而可以监控Hadoop集群、队列使用情况、jvm情况等。不过如果你想监控Job的详细详细, 这个可能就满足不了。下篇文章将介绍如何获取Job的信息。

本博客文章除特别声明, 全部都是原创!

禁止个人和公司转载本文、谢谢理解: 过往记忆 (<https://www.iteblog.com/>)

本文链接: 【Hadoop集群监控: jmx信息获取】 (<https://www.iteblog.com/archives/1694.html>)

♡ 喜欢 (10)

赏

↪ 分享 (0)