

图文 09 生产经验：如何为生产环境中的数据库部署监控系统？

939 人次阅读 2020-02-05 08:01:17

- 返回
- 前进
- 重新加载
- 打印

详情 评论

生产经验：如何为生产环境中的数据库部署监控系统？

如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑

如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《MySQL专栏付费用户如何加群》（购买后可见）



进店逛

1、生产环境的数据库可不能裸奔啊！

之前我们已经给大家讲解完了数据库的压测相关的知识，想必大家只要利用我们讲解的知识，在自己的公司里，哪怕DBA团队、QA团队都没法给你提供专业的数据库压测技术支持，但是大家手上拿到一个数据库之后，其实自己也可以通过各种工具和命令，非常好的完成一台机器上的数据库的压测了。

你应该可以心里非常有数，一台什么样配置的机器，部署了一个数据库之后，利用sysbench构造了多少个表和数据量，然后模拟了多少个线程压测的时候，机器的各项硬件负载在可以接受的范围内时，数据库的QPS和TPS可以压测到多高。

这个时候你大致就明白你的数据库在高峰时期最多可以让他去承受多少QPS和TPS了。

但是搞定压测之后，难道大家就想直接开始开发你的Java系统？直接让你的系统连接到MySQL上去执行各种CRUD的SQL语句？然后接着就开始拼命写各种Java代码和SQL语句，写好之后就找QA进行测试，然后部署到线上生产环境，接着就万事大吉了，不管数据库了？

这种做法可能目前很多公司和团队都是这样做的，但是如果你仅仅是这么搞是绝对不行的。因为实际上我们需要对线上系统进行完善的监控，不光是对你开发的Java系统进行监控，还得对你的数据库进行监控，包括对CPU、内存、网络、磁盘IO、慢查询、QPS、TPS的监控。

因为如果你不对你的数据库做任何监控，那么有可能你的数据库CPU负载已经很高了，或者磁盘IO已经达到极限了，你都不知道，结果你还是一如既往的运行你的Java系统，有一天可能你的数据库突然挂了你都没反应过来！

所以今天我们就带着大家来一步步搭建一下生产环境数据库的可视化监控平台，我们会基于Prometheus+Grafana来搭建。

当然在公司里，如果要针对数据库搭建一个统一的可视化监控平台，这个活儿往往是DBA团队负责的，但是不管如何，我们这里也要对这个数据库可视化监控的技术有一定的了解。

2、简单介绍一下Prometheus和Grafana是什么

我们先给大家简单介绍一下Prometheus和Grafana两个系统分别是什么。

简单来说，Prometheus其实就是一个监控数据采集和存储系统，他可以利用监控数据采集组件（比如mysql_exporter）从你指定的MySQL数据库中采集他需要的监控数据，然后他自己有一个时序数据库，他会把采集到的监控数据放入自己的时序数据库中，其实本质就是存储在磁盘文件里。

相关频道



从零开始
MySQL 进阶与调优
已更新3

我们采集到了MySQL的监控数据还不够，现在我们还要去看这些数据组成的一些报表，所以此时就需要使用Grafana了，Grafana就是一个可视化的监控数据展示系统，他可以把Prometheus采集到的大量的MySQL监控数据展示成各种精美的报表，让我们可以直观的看到MySQL的监控情况。

其实不光是对数据库监控可以采用Prometheus+Grafana的组合，对你开发出来的各种Java系统、中间件系统，都可以使用这套组合去进行可视化的监控，无非就是让Prometheus去采集你的监控数据，然后用Grafana展示成报表而已。

3、安装和启动Prometheus

之前给大家说过，让大家自己准备一个linux机器，如果你是windows笔记本电脑，可以自己装一个linux虚拟机。我们就基于一台linux机器来部署Prometheus和Grafana，至于MySQL的安装，这个非常的简单，大家在网上很容易搜索到。

首先大家需要下载3个压缩包，在下面链接：

<http://cactifans.hi-www.com/prometheus/>

大家可以下载到下面两个压缩包，这里prometheus就是用来部署监控系统自己的，然后node_exporter是用来采集MySQL数据库所在机器的CPU、内存、网络、磁盘之类的监控数据的：

prometheus-2.1.0.linux-amd64.tar.gz

node_exporter-0.15.2.linux-amd64.tar.gz

接着大家可以通过下面的链接下载第三个压缩包：mysqld_exporter-0.10.0.linux-amd64.tar.gz，这个mysqld_exporter就是用来采集MySQL数据库自己的一些监控数据的，比如SQL性能、连接数量之类的：

https://github.com/prometheus/mysqld_exporter/releases/download/v0.10.0/mysqld_exporter-0.10.0.linux-amd64.tar.gz

接着需要解压缩上面的几个包，参照我如下的命令来做就可以了：

```
mkdir /data
```

```
mkdir /root
```

```
tar xvf prometheus-2.1.0.linux-amd64.tar -C /data
```

```
tar xf node_exporter-0.15.2.linux-amd64.tar -C /root
```

```
tar xf mysqld_exporter-0.10.0.linux-amd64.tar.gz -C /root
```

```
cd /data
```

```
mv prometheus-2.1.0.linux-amd64/ prometheus
```

```
cd /prometheus
```

vi prometheus.yml，接下来修改prometheus的配置文件，其实主要是在scrape_configs下面加入一大段自定义的配置，因为他需要去采集MySQL数据库本身和MySQL所在机器的监控数据：

```
scrape_configs:
```

```
- file_sd_configs:
```

```
- files:
```

```
- host.yml
```

```
job_name: Host
```

```
metrics_path: /metrics
```

```
relabel_configs:
```

```
- source_labels: [__address__]
```

```
  regex: (.*)
```

```
  target_label: instance
```

```
  replacement: $1
```

```
- source_labels: [__address__]
```

```
  regex: (.*)
```

```
  target_label: __address__
```

```
  replacement: $1:9100
```

```
- file_sd_configs:
```

认准一手QQ3195303913微信wxywd8

```
- files:
  - mysql.yml
job_name: MySQL
metrics_path: /metrics
relabel_configs:
- source_labels: [__address__]
  regex: (.*)
  target_label: instance
  replacement: $1
- source_labels: [__address__]
  regex: (.*)
  target_label: __address__
  replacement: $1:9104

- job_name: prometheus
  static_configs:
  - targets:
    - localhost: 9090
```

上面的配置文件写好之后，就可以启动Prometheus了，不过大家仔细看几遍上面的配置信息，因为我在写文章的时候不太方便，都是直接手敲出来的，可能会有少数配置错误，如果大家有发现配置文件错误的地方，及时在后台评论区告诉我。

接着必须要在/data/prometheus目录中，去执行启动命令：

/data/prometheus/prometheus --storage.tsdb.retention=30d &，这里的30d是说你的监控数据保留30天的。启动之后，就可以在浏览器中访问9090端口号去查看prometheus的主页了。

因为我们部署和安装Prometheus和Grafana的过程比较多，所以拆分为两篇文章，今天同步更新的第二篇文章里，会把剩余的Grafana的安装部署过程，以及监控配置和采集的过程都讲完。

认准一手QQ195303913微信wxywd8

专栏版权归公众账号狸猫技术窝所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

[《从零开始带你成为消息中间件实战高手》](#)
[《21天互联网Java进阶面试训练营》（分布式篇）](#)
[《互联网Java工程师面试突击》（第1季）](#)
[《互联网Java工程师面试突击》（第3季）](#)
[《从零开始带你成为JVM实战高手》](#)